Mahidol University
Faculty of Medicine Ramathibodi Hospital
Department of Clinical Epidemiology and Biostatistics

# Using a neural network-based feature extraction method to facilitate citation screening for systematic reviews

**Presented by**

**Mr. Phongphat Wiwatthanasetthakarn (G6236262)**

**Master of Science Program in Data Science for Health Care**

**(International Program)**

**Faculty of Medicine Ramathibodi Hospital**

**Mahidol University**

Expert Systems with Applications: X 6 (2020) 100030

Contents lists available at ScienceDirect

Expert Systems with Applications: X

journal homepage: www.elsevier.com/locate/eswax


# Using a neural network-based feature extraction method to facilitate citation screening for systematic reviews

Georgios Kontonatsios [a,*], Sally Spencer [b], Peter Matthew [a], Ioannis Korkontzelos [a]

[a] *Department of Computer Science, Edge Hill University, United Kingdom*
[b] *Faculty of Health and Social Care, Edge Hill University, United Kingdom*

# Topics

- Research question & objectives
- Background and rational
- Fundamental concepts of model development
- Methodology
- Experiments
- Contributions

# Research question & objectives

## Research question

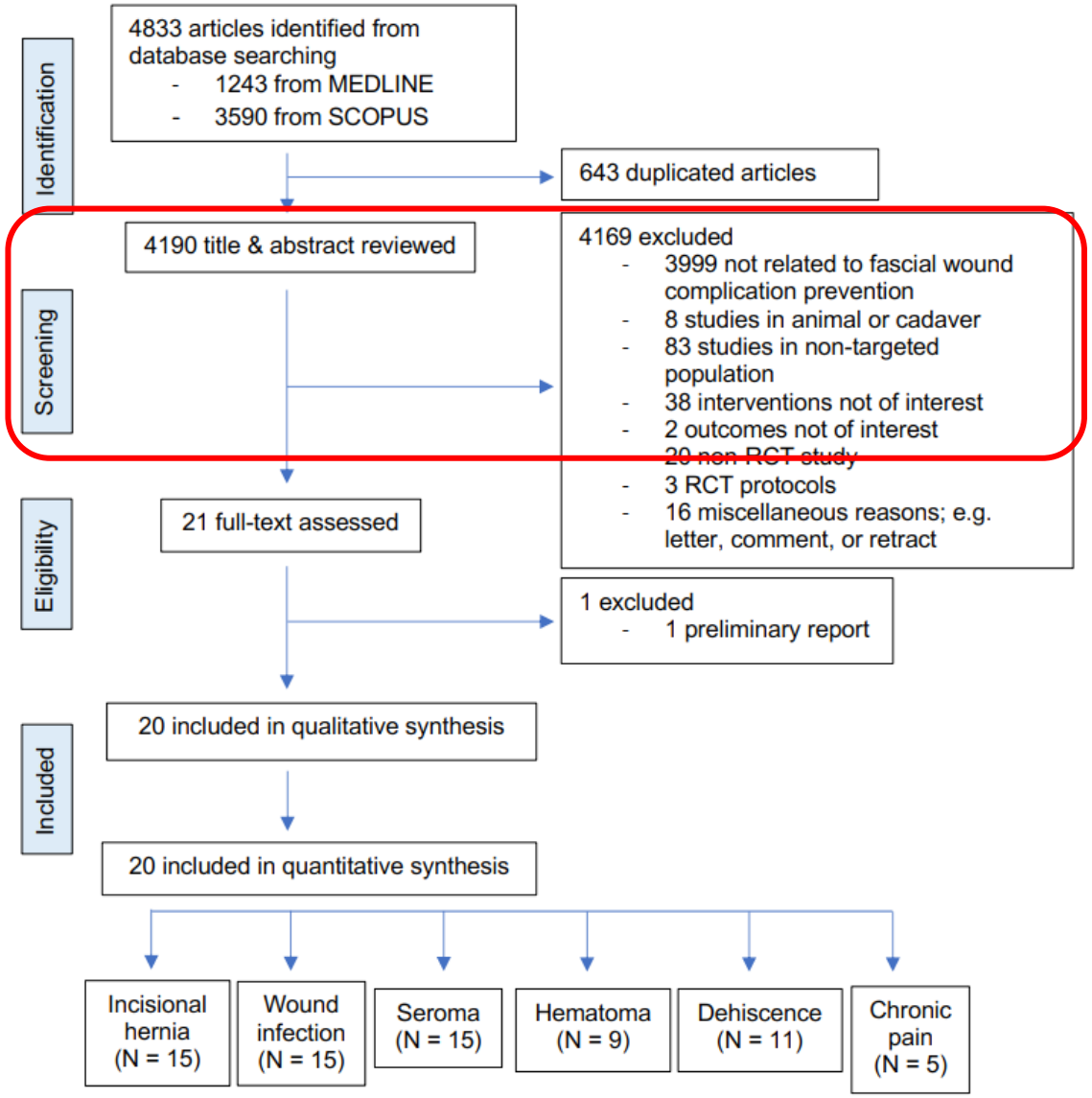
- How well a neural network-based feature extraction method to facilitate citation screening for systematic reviews when compare to others existing method?

## Objectives

- Propose an automatic text classification method to accelerate the citation screening process of systematic reviews by employing a neural network-based feature extraction method.
- Assess the performance of proposed feature extraction method and compare with different baseline feature extraction methods using linear SVM text classifier.

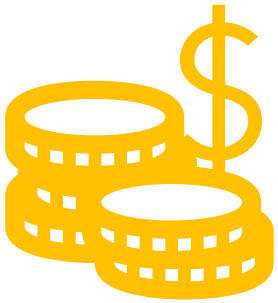
# Background and rational

## PRISMA flow




Time

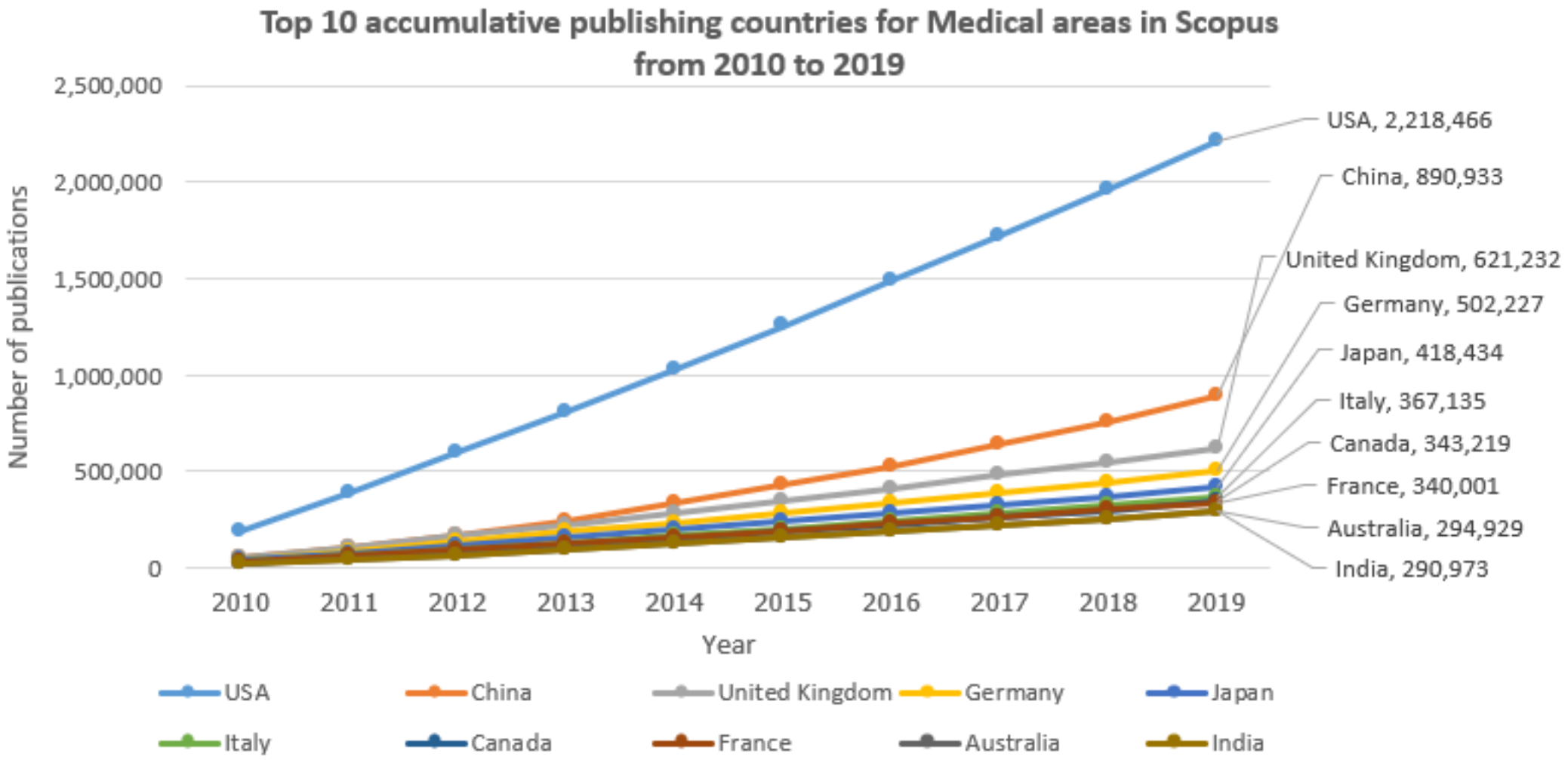
Workload



Cost

# Background and rational

## Top 10 accumulative publishing countries for Medical areas in Scopus from 2010 to 2019



# Background and rational

## Eligible rate

**Table 1**
23 publicly available review datasets used in the experiments of this paper.

| Source | Dataset | # citations | (%) eligible citations | Bibliographic metadata |
|---|---|---|---|---|
| Clinical (Wallace et al., 2010) | COPD | 1606 | 12.2 | ✗ |
| | Proton Beam | 4751 | 5.1 | ✗ |
| | Micro Nutrients | 4010 | 6.4 | ✗ |
| | ACEInhibitors | 2544 | 1.6 | ✓ |
| | ADHD | 851 | 2.4 | ✓ |
| | Antihistamines | 310 | 5.2 | ✓ |
| | Atypical Antipsychotics | 1120 | 13.0 | ✓ |
| | Beta Blockers | 2072 | 2.0 | ✓ |
| | Calcium Channel Blockers | 1218 | 8.2 | ✓ |
| | Estrogens | 368 | 21.7 | ✓ |
| Drug (Cohen et al., 2006) | NSAIDs | 393 | 10.4 | ✓ |
| | Opioids | 1915 | 0.8 | ✓ |
| | Oral Hypoglycemics | 503 | 27.0 | ✓ |
| | Proton PumpInhibitors | 1333 | 3.8 | ✓ |
| | Skeletal Muscle Relaxants | 1643 | 0.5 | ✓ |
| | Statins | 3465 | 2.5 | ✓ |
| | Triptans | 671 | 3.6 | ✓ |
| | Urinary Incontinence | 327 | 12.2 | ✓ |
| | PFOA/PFOS | 6330 | 1.5 | ✓ |
| SWIFT (Howard et al., 2016) | Bisphenol A (BPA) | 7699 | 1.4 | ✓ |
| | Transgenerational | 48,637 | 1.6 | ✓ |
| | Fluoride and neurotoxicity | 4479 | 1.1 | ✗ |
| | Neuropathic pain | 29,207 | 17.2 | ✗ |

# Fundamental concepts of model development

- Existing feature extractions
- Neural network
- Autoencoders (AE)
- Denoising Autoencoders (DAE)
- Support vector machines

# Existing feature extractions

## Process


## Example

Did not like a good movie?

- Convert to lower case characters
- Remove punctuations
- Remove stop words
- Stemming
- Lemmatization

- Bag-of-Words
- TF-IDF

| good | movie | not | a | did | like |
|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 |

- Logistic Regression
- Support Vector Machine
- Decision Tree
- Random Forest

- positive
- negative
- neutral

# Existing feature extractions

## Bag-of-Words

| | the | red | dog | cat | eats | food |
|---|---|---|---|---|---|---|
| 1. the red dog → | 1 | 1 | 1 | 0 | 0 | 0 |
| 2. cat eats dog → | 0 | 0 | 1 | 1 | 1 | 0 |
| 3. dog eats food → | 0 | 0 | 1 | 0 | 1 | 1 |
| 4. red cat eats → | 0 | 1 | 0 | 1 | 1 | 0 |

# Existing feature extractions

## Term frequency - inverse document frequency (TF-IDF)

$\texttt{tf}(t, d)$

| | blue | bright | can | see | shining | sky | sun | today |
|---|---|---|---|---|---|---|---|---|
| 1 | 1/2 | 0 | 0 | 0 | 0 | 1/2 | 0 | 0 |
| 2 | 0 | 1/3 | 0 | 0 | 0 | 0 | 1/3 | 1/3 |
| 3 | 0 | 1/3 | 0 | 0 | 0 | 1/3 | 1/3 | 0 |
| 4 | 0 | 1/6 | 1/6 | 1/6 | 1/6 | 0 | 1/3 | 0 |

x

$\texttt{idf}(t, D)$

| | blue | bright | can | see | shining | sky | sun | today |
|---|---|---|---|---|---|---|---|---|
| | 0.602 | 0.125 | 0.602 | 0.602 | 0.602 | 0.301 | 0.125 | 0.602 |

- TF-IDF: Multiply TF and IDF scores, use to rank importance of words within documents
  - Most important word for each document is highlighted

$\texttt{tfidf}(t, d, D) = \texttt{tf}(t, d) \cdot \texttt{idf}(t, D)$

| | blue | bright | can | see | shining | sky | sun | today |
|---|---|---|---|---|---|---|---|---|
| 1 | **0.301** | 0 | 0 | 0 | 0 | 0.151 | 0 | 0 |
| 2 | 0 | 0.0417 | 0 | 0 | 0 | 0 | 0.0417 | **0.201** |
| 3 | 0 | 0.0417 | 0 | 0 | 0 | **0.100** | 0.0417 | 0 |
| 4 | 0 | 0.0209 | **0.100** | **0.100** | **0.100** | 0 | 0.0417 | 0 |

# Existing feature extractions

## TF-IDF

**Variants of term frequency (tf) weight**

| weighting scheme | tf weight |
|---|---|
| binary | $0, 1$ |
| raw count | $f_{t,d}$ |
| term frequency | $f_{t,d} \Big/ \sum_{t' \in d} f_{t',d}$ |
| log normalization | $\log(1 + f_{t,d})$ |
| double normalization 0.5 | $0.5 + 0.5 \cdot \frac{f_{t,d}}{\max_{\{t' \in d\}} f_{t',d}}$ |
| double normalization K | $K + (1 - K)\frac{f_{t,d}}{\max_{\{t' \in d\}} f_{t',d}}$ |

**Variants of inverse document frequency (idf) weight**

| weighting scheme | idf weight ($n_t = \lvert\{d \in D : t \in d\}\rvert$) |
|---|---|
| unary | 1 |
| inverse document frequency | $\log \frac{N}{n_t} = -\log \frac{n_t}{N}$ |
| inverse document frequency smooth | $\log\left(\frac{N}{1+n_t}\right) + 1$ |
| inverse document frequency max | $\log\left(\frac{\max_{\{t' \in d\}} n_{t'}}{1+n_t}\right)$ |
| probabilistic inverse document frequency | $\log \frac{N - n_t}{n_t}$ |

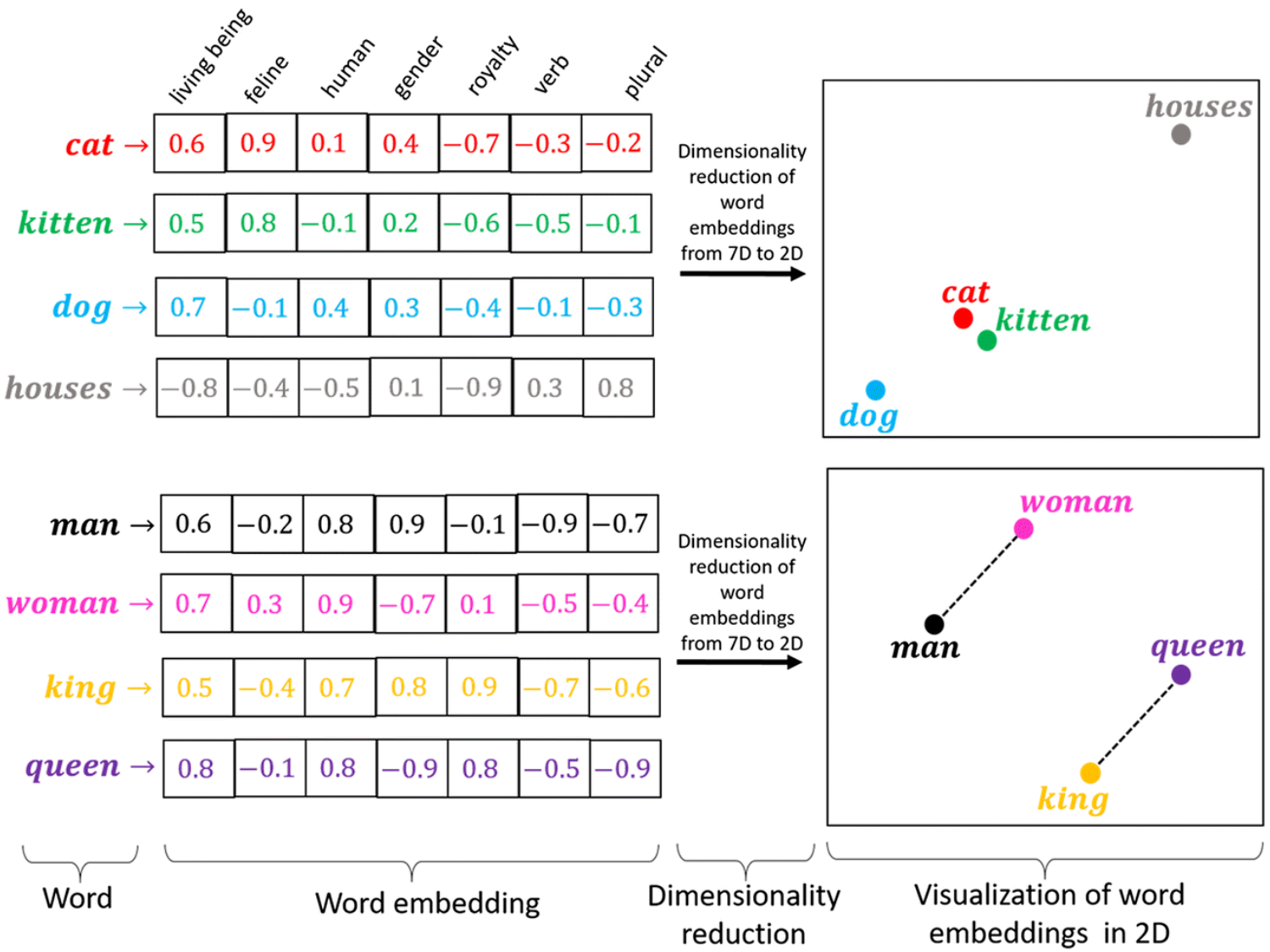
**Recommended tf–idf weighting schemes**

| weighting scheme | document term weight | query term weight |
|---|---|---|
| 1 | $f_{t,d} \cdot \log \frac{N}{n_t}$ | $\left(0.5 + 0.5\frac{f_{t,q}}{\max_t f_{t,q}}\right) \cdot \log \frac{N}{n_t}$ |
| 2 | $\log(1 + f_{t,d})$ | $\log\left(1 + \frac{N}{n_t}\right)$ |
| 3 | $(1 + \log f_{t,d}) \cdot \log \frac{N}{n_t}$ | $(1 + \log f_{t,q}) \cdot \log \frac{N}{n_t}$ |

Source: https://en.wikipedia.org/wiki/Tf%E2%80%93idf

# Existing feature extractions

## Word Embedding

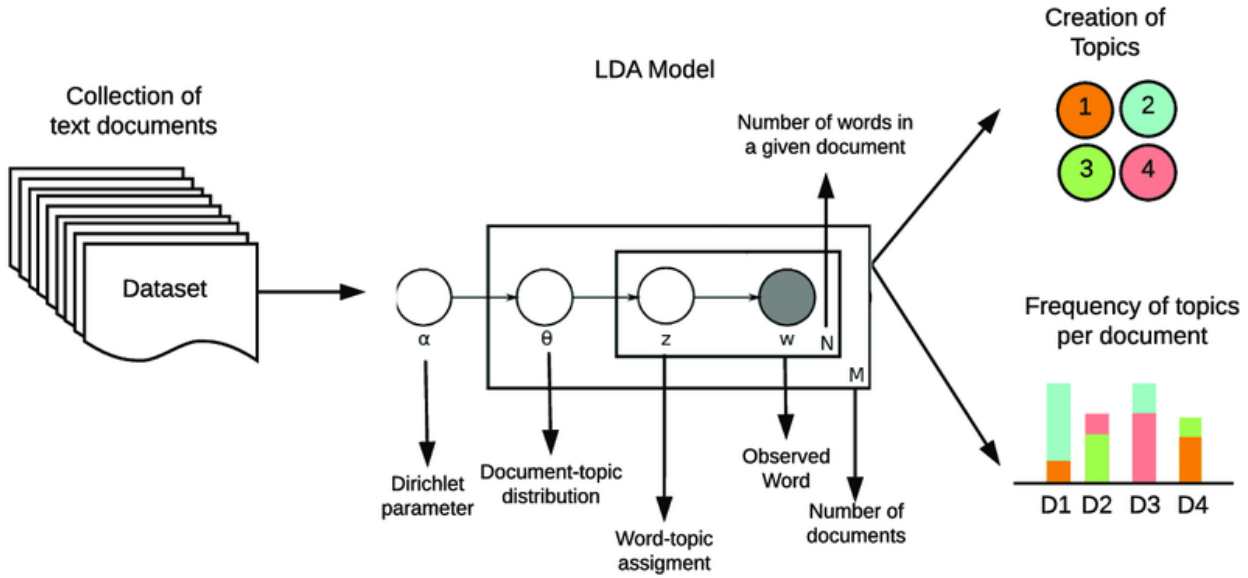


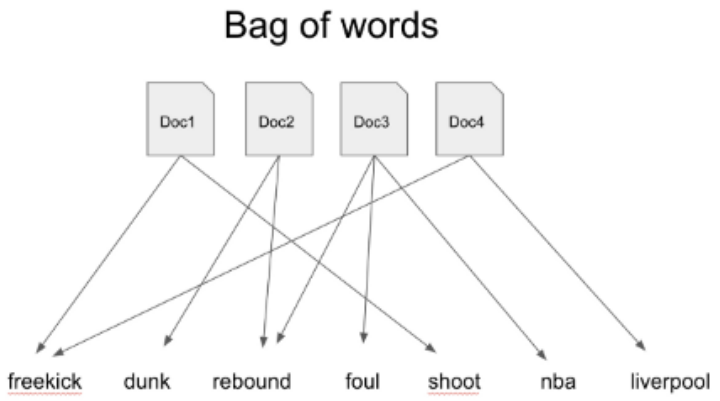

**Word embedding**

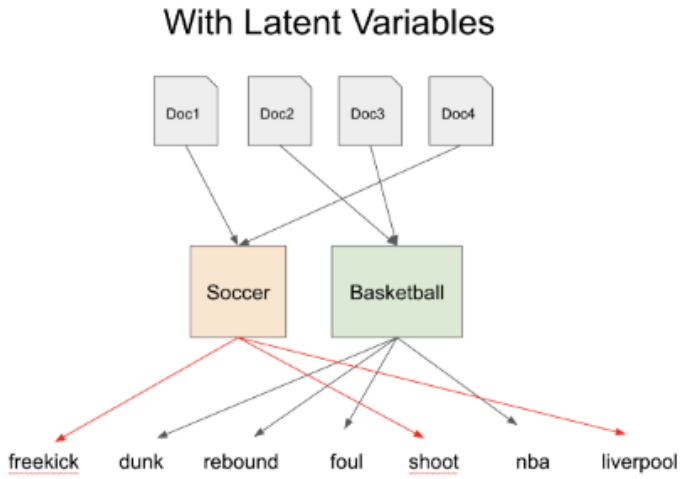
- A term used for the representation of words for text analysis.
- Typically, in the form of a real-valued vector that encodes the meaning of the word.
- The words that are closer in the vector space are expected to be similar in meaning.

# Existing feature extractions

## Latent Dirichlet Allocation (LDA)




- A popular topic modeling technique to extract topics from a given corpus.
- The term latent conveys something that exists but is not yet developed.
- In other words, latent means hidden or concealed.
- LDA breaks the corpus document word into lower-dimensional matrices.

Source:https://www.analyticsvidhya.com/blog/2021/06/part-2-topic-modeling-and-latent-dirichlet-allocation-lda-using-gensim-and-sklearn/

# Existing feature extractions

## Singular Value Decomposition (SVD)

- SVD is a technique from linear algebra that can be used to automatically perform dimensionality reduction.
- Dimensionality reduction involves reducing the number of input variables or columns in modeling data.
- Most popular technique for dimensionality reduction when data is sparse.
- Sparse data refers to rows of data where many of the values are zero.

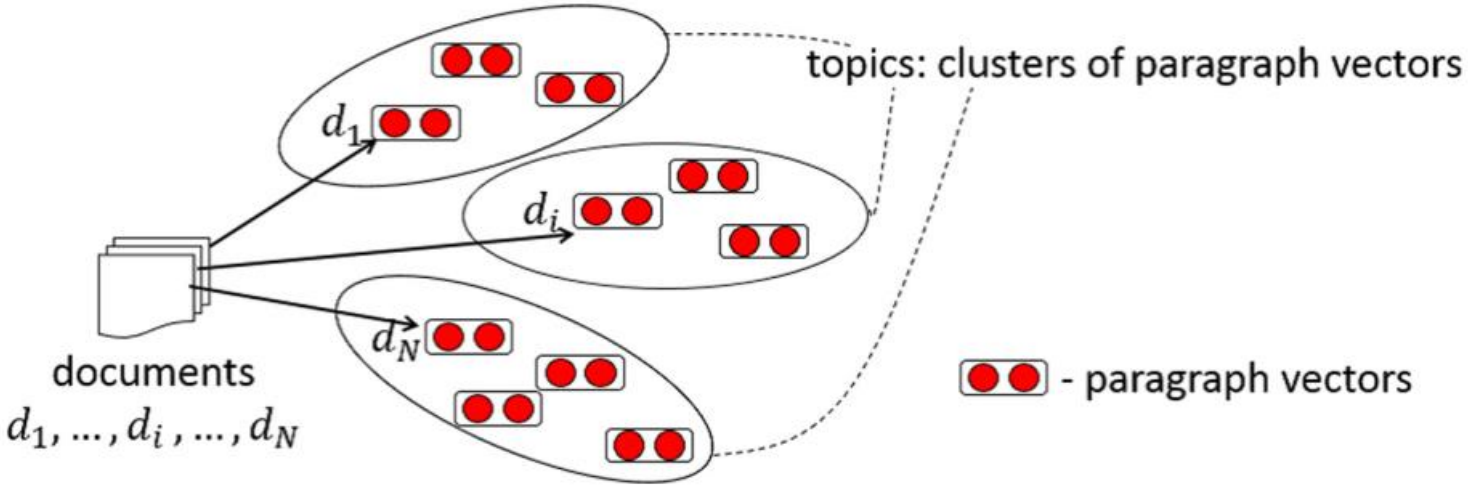


Source:https://machinelearningmastery.com/singular-value-decomposition-for-dimensionality-reduction-in-python/

# Existing feature extractions

## Paragraph Vector (PV)

- Paragraph Vector is one of such algorithms, which extends the **word2vec** algorithm by considering the paragraph as an additional word.
- Word vectors represent only words, paragraph vectors represent phrases, sentences, paragraphs and documents of arbitrary length.
- It proved to be successful in several applications, including document classification and document similarity calculation.

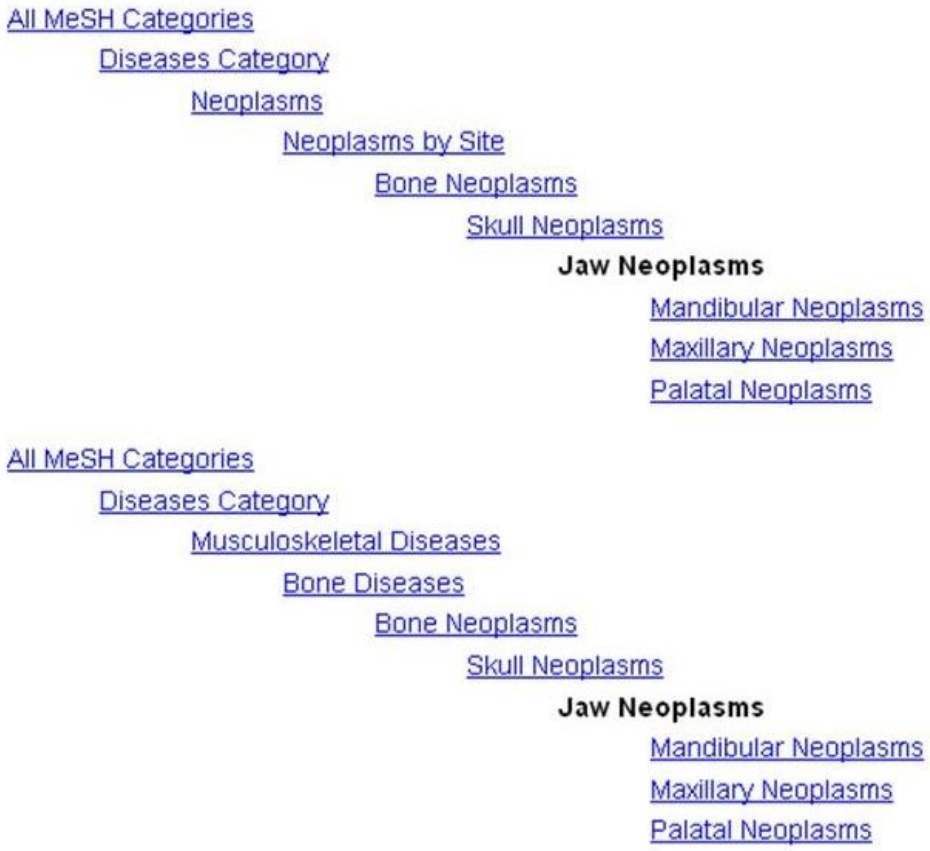



Source: https://ieeexplore.ieee.org/document/8653834

# Existing feature extractions

## Medical Subject Headings (MeSH)

For instance, if you want to search cancer of the jaw

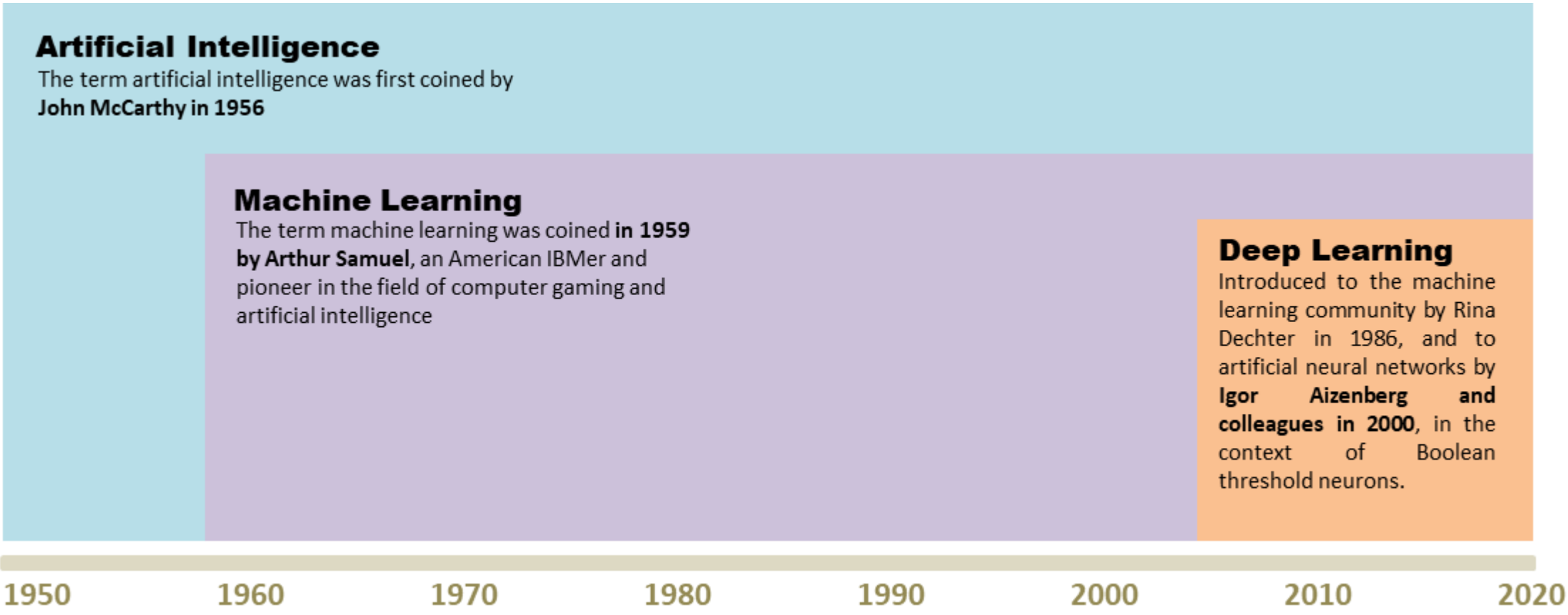



- MeSH tags are single word or multi-word keywords that are manually assigned to every citation indexed by the Medline bibliographic database.
- MeSH tags aim at summarising the textual content of citations using a set of descriptive keywords.
- Considering that MeSH keywords may not always appear in the title or in the abstract of a citation.
- MeSH-based features can potentially provide complimentary information to BoW features.

Source: https://research.library.gsu.edu/c.php?g=115556&p=753156

# Neural network

## AI, ML, DL






Source: https://datascience.foundation/sciencewhitepaper/natural-language-processing-nlp-simplified-a-step-by-step-guide

# Neural network

## AI, ML, DL

### Artificial Intelligence

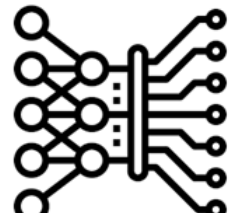
In short - incorporating human intelligence to machines.

Whenever a machine completes tasks based on a set of stipulated rules that solve problems (algorithms), such an "intelligent" behavior can be termed as artificial intelligence.

### Machine Learning

As the name suggests, machine learning can be loosely interpreted to mean empowering computer systems with the ability to "learn".

The intention of ML is to enable machines to learn by themselves using the provided data and make accurate predictions. It is the field of study that gives computers the capability to learn without being explicitly programmed. ML is a subset of artificial intelligence; in fact, it's simply a technique for realizing AI.



### Deep Learning

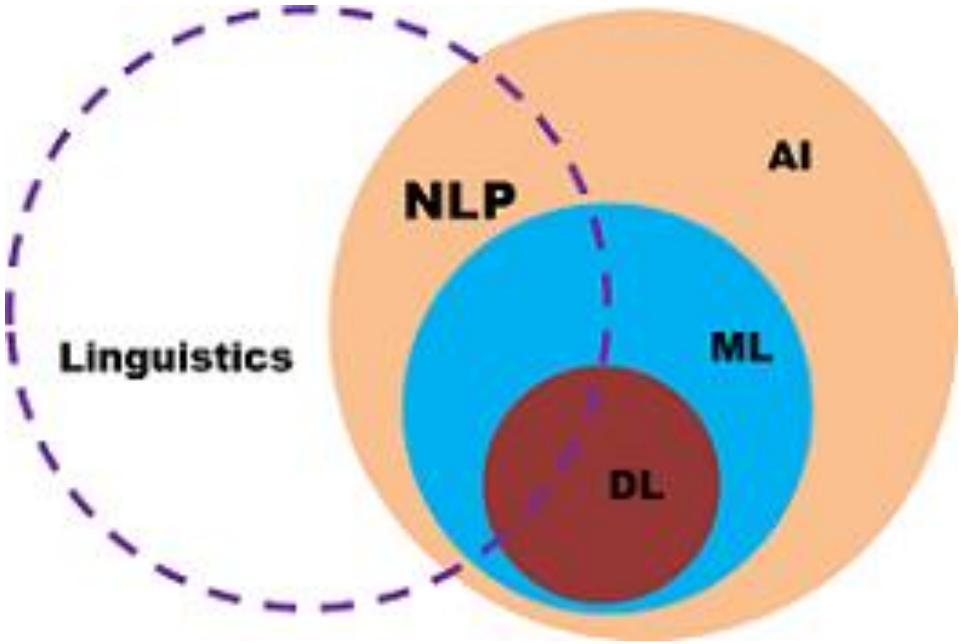
DL is the next evolution of machine learning. DL algorithms are roughly inspired by the information processing patterns found in the human brain. Just like we use our brains to identify patterns and classify various types of information, deep learning algorithms can be taught to accomplish the same tasks for machines.



Source: https://datascience.foundation/sciencewhitepaper/natural-language-processing-nlp-simplified-a-step-by-step-guide

# Neural network

## AI, ML, DL, NLP



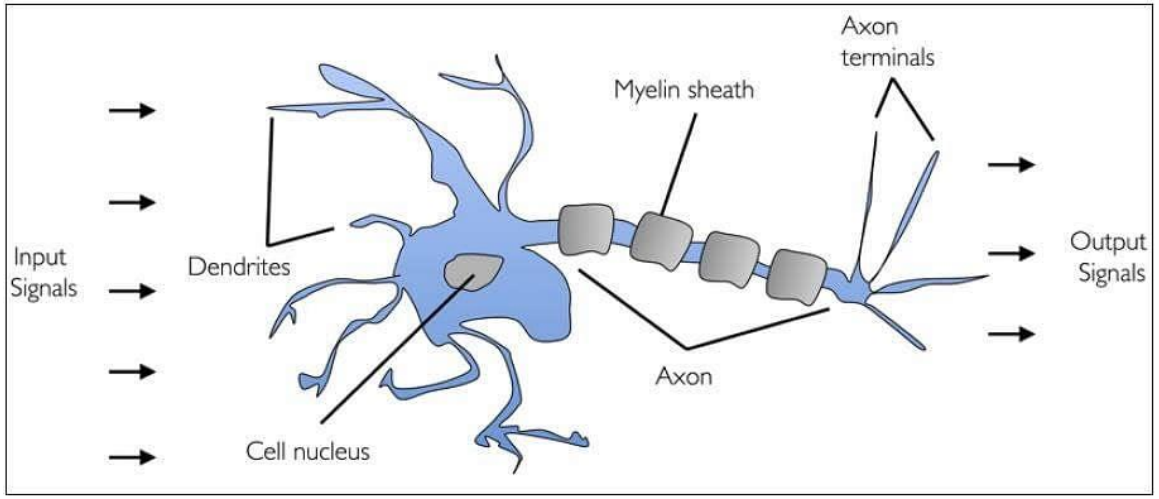

**Natural language processing (NLP)**

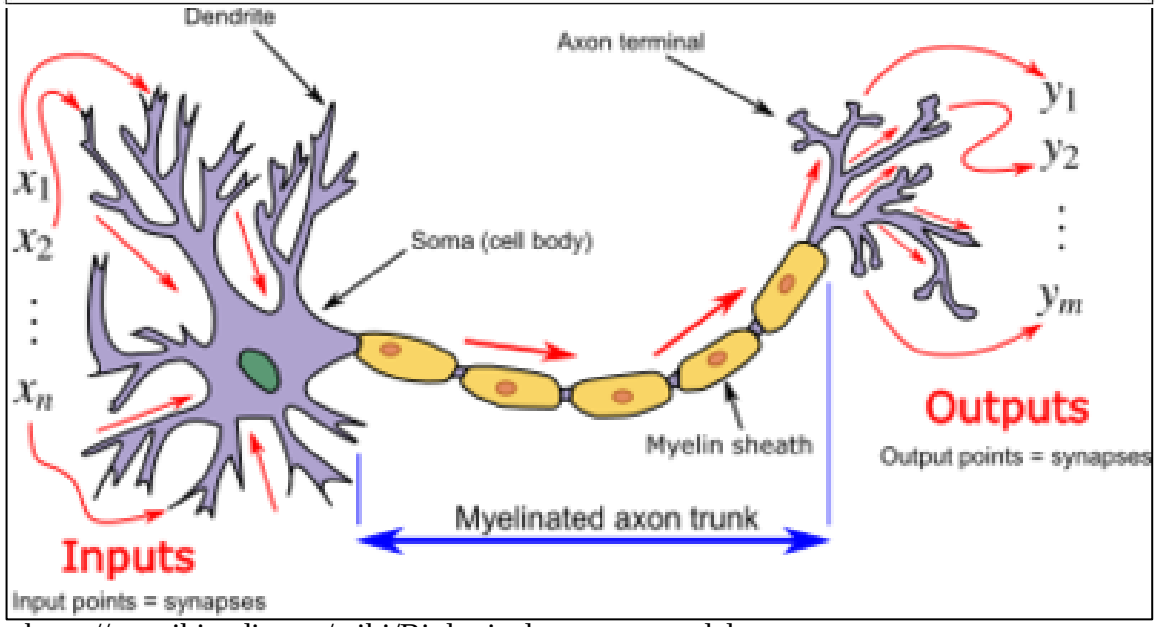
- A subfield of linguistics, computer science, and artificial intelligence.
- Concerned with the interactions between computers and human language.
- In particular how to program computers to process and analyze large amounts of natural language data.
- The goal is a computer capable of "understanding" the contents of documents, including the contextual nuances of the language within them.

Source: https://datascience.foundation/sciencewhitepaper/natural-language-processing-nlp-simplified-a-step-by-step-guide
https://en.wikipedia.org/wiki/Natural_language_processing

# Neural network

## Neural network in human brain




| Biological Neuron | Artificial Neuron |
|---|---|
| Cell Nucleus (Soma) | Node |
| Dendrites | Input |
| Synapse | Weights or interconnections |
| Axon | Output |

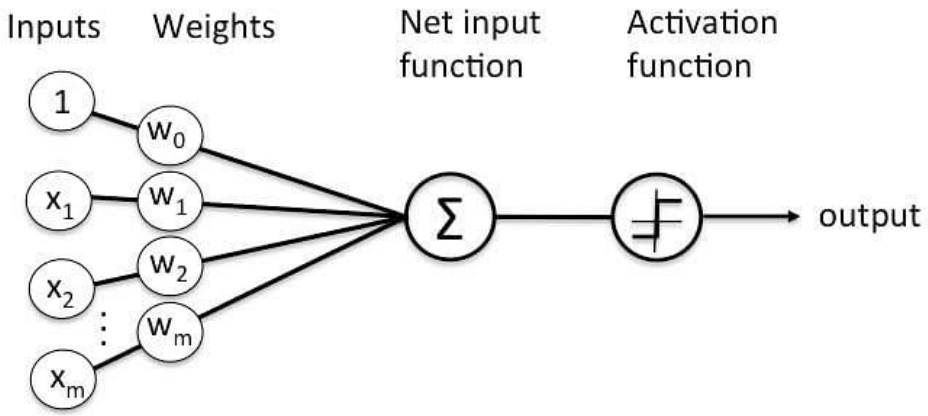
Source: https://en.wikipedia.org/wiki/Biological_neuron_model
https://www.simplilearn.com/tutorials/deep-learning-tutorial/perceptron#:~:text=A%20Perceptron%20is%20a%20neural,value%20%E2%80%9Df(x).

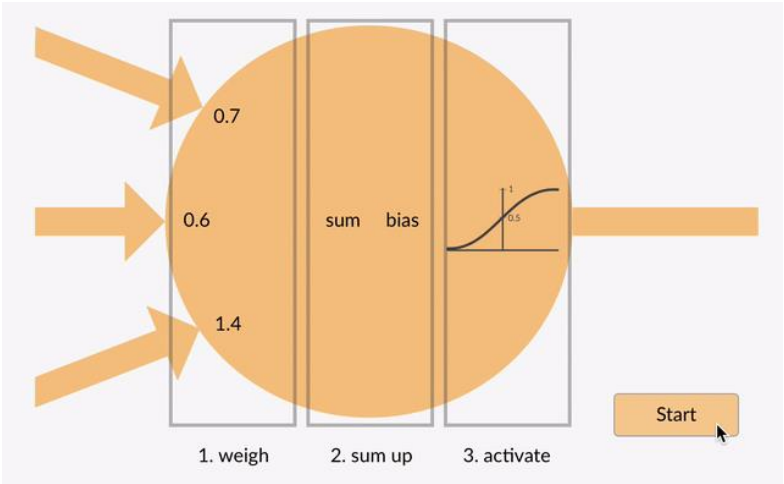
# Neural network

## Perceptron

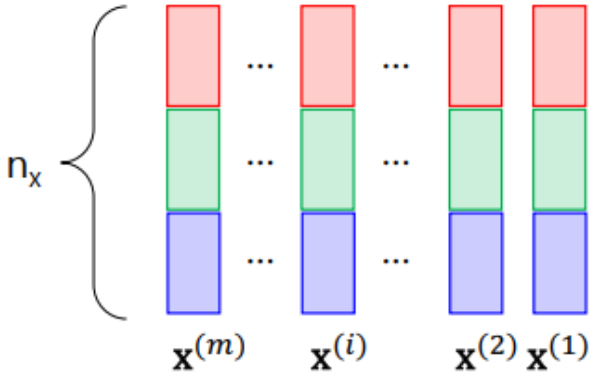
**The smallest unit of neural network**

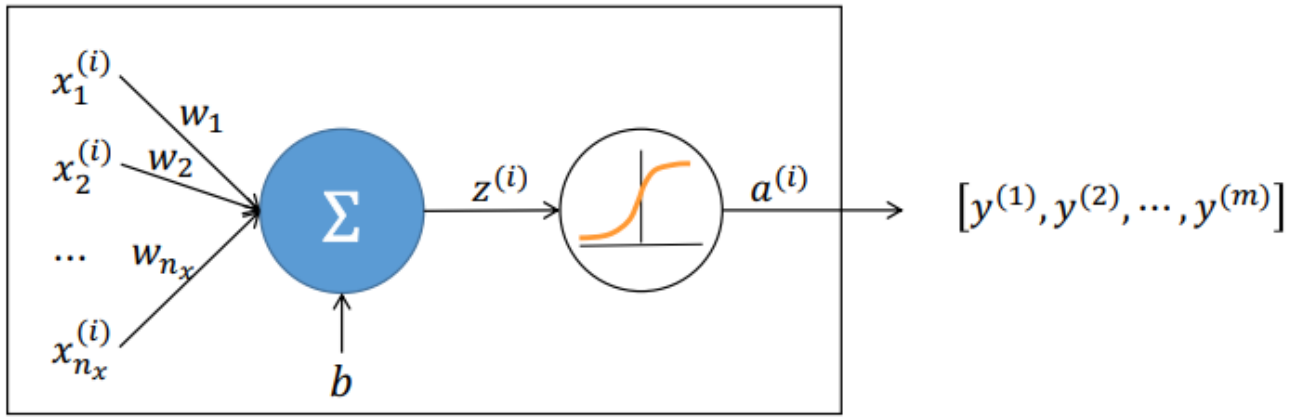
Perceptron was introduced by Frank Rosenblatt in 1957.




**Automatically extract the features**



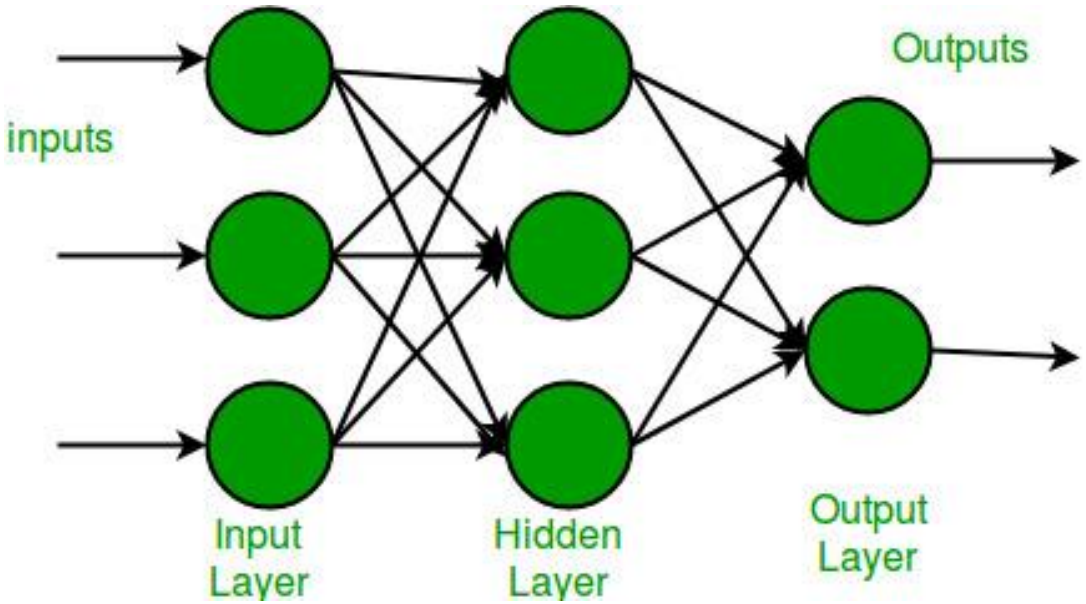




Source: RADS 602: Data Mining and Machine Learning, Ratchainant Thammasudjarit (Ph.D.)
https://www.simplilearn.com/tutorials/deep-learning-tutorial/perceptron#:~:text=A%20Perceptron%20is%20a%20neural,value%20%E2%80%9Df(x).

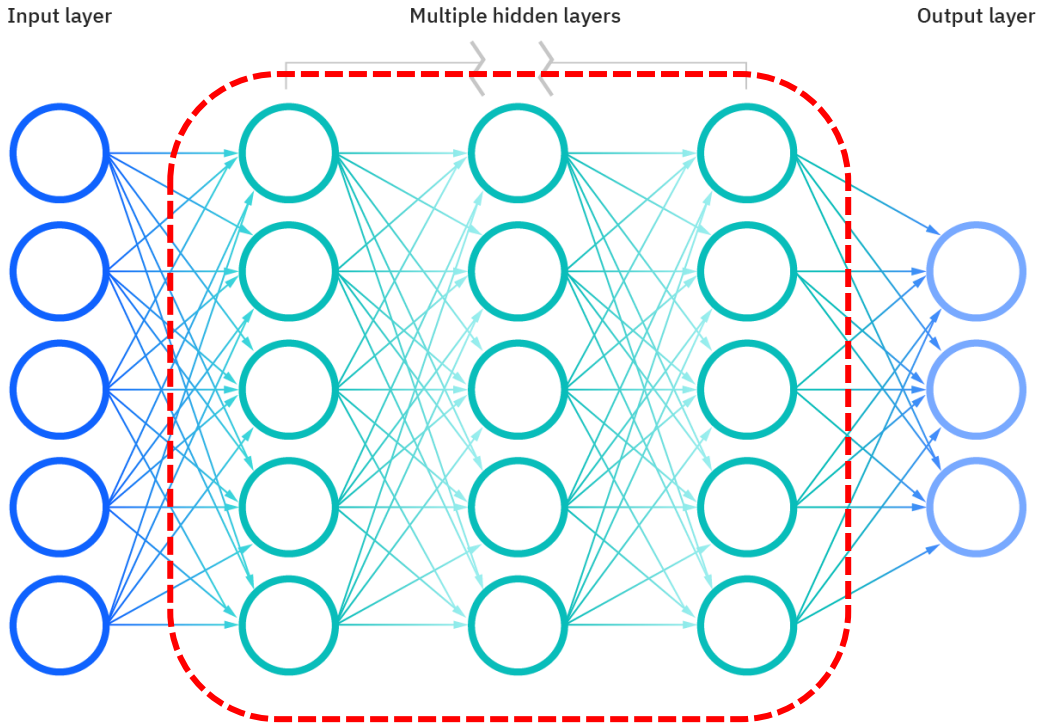
# Neural network

## Artificial neural network vs Deep neural network

### Artificial neural network



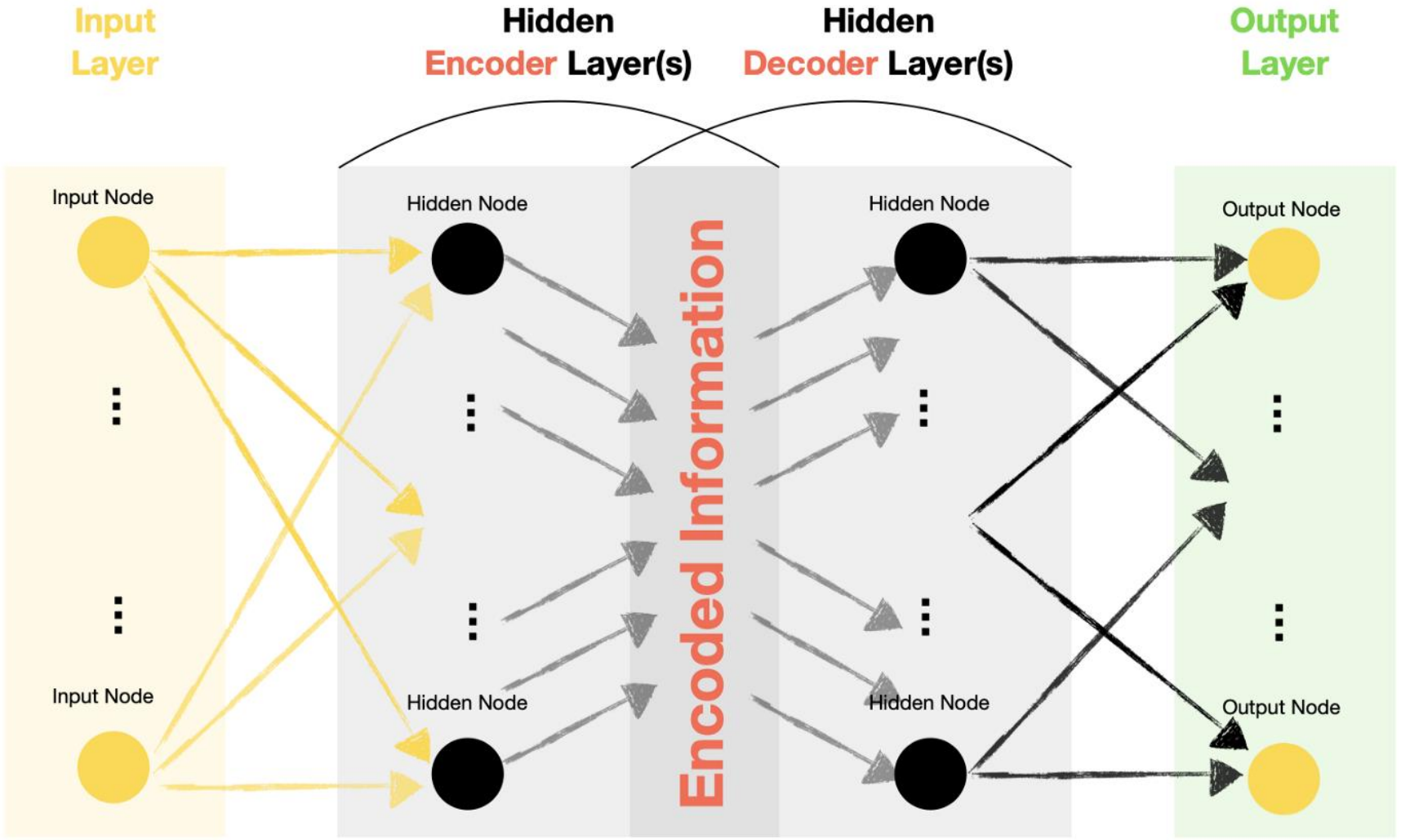

### Deep neural network




Source: https://www.ibm.com/cloud/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks
https://www.geeksforgeeks.org/difference-between-a-neural-network-and-a-deep-learning-system/#:~:text=Deep%20Learning%20is%20associated%20with,help%20of%20the%20various%20connections.

# Neural network

## Autoencoders (AE)




Source: https://towardsdatascience.com/autoencoders-ae-a-smart-way-to-process-your-data-using-unsupervised-neural-networks-9661f93a8509

# Neural network

## Autoencoders (AE)

- A type of artificial neural network used to learn efficient coding of unlabeled data (unsupervised learning)
- The autoencoder learns a representation (encoding) for a set of data, typically for dimensionality reduction, by training the network to ignore insignificant data (“noise”).
- Can be applied for a variety of tasks such as:
  - Dimensionality reduction
  - Feature extraction
  - Denoising of data/images
  - Imputing missing data

Source: https://en.wikipedia.org/wiki/Autoencoder
https://towardsdatascience.com/autoencoders-ae-a-smart-way-to-process-your-data-using-unsupervised-neural-networks-9661f93a8509

# Neural network

## Autoencoders (AE)

### Types of Autoencoders:

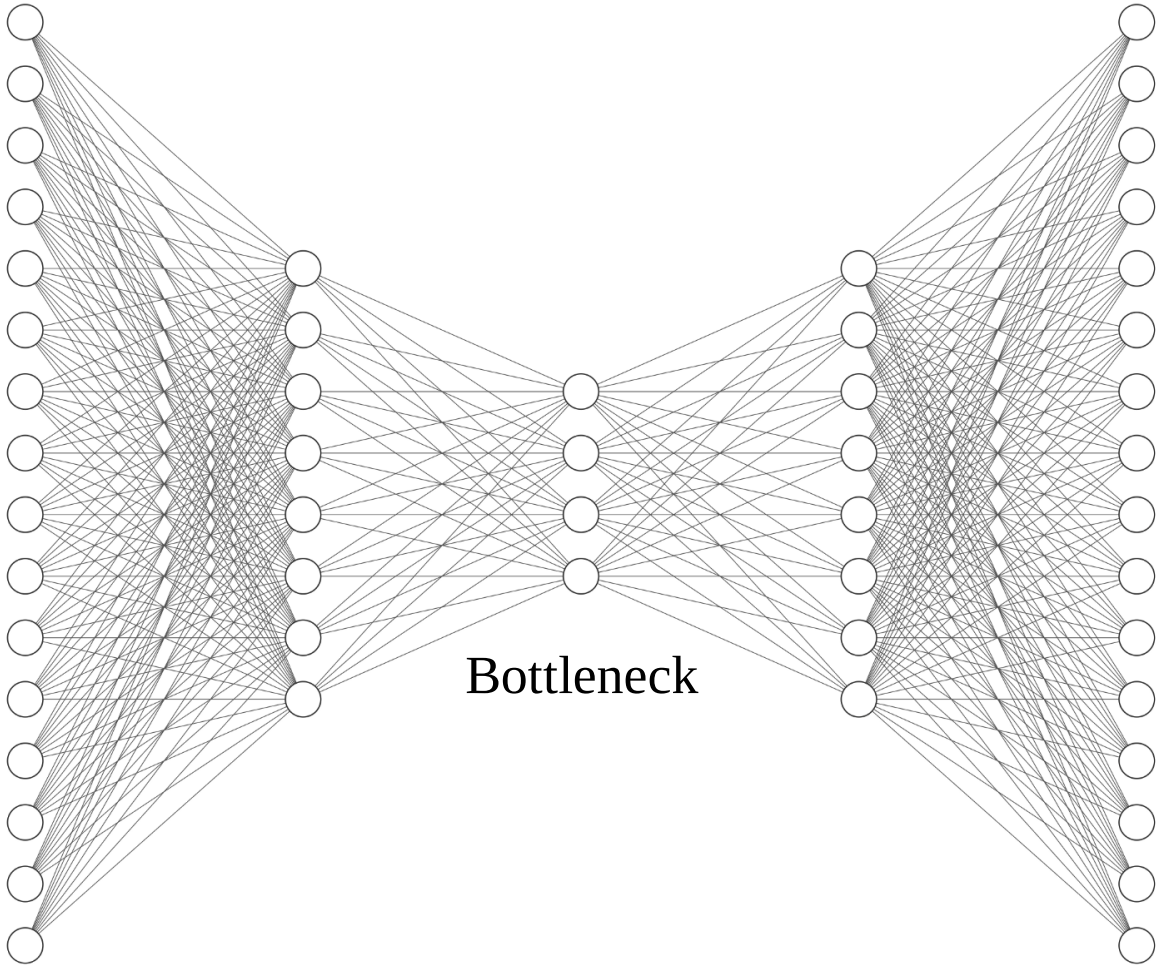
1. **Undercomplete Autoencoder (the focus of this article):**
   - Has fewer nodes (dimensions) in the middle compared to Input and Output layers. In such setups, we tend to call the middle layer a "bottleneck."

2. **Overcomplete Autoencoder:**
   - Has more nodes (dimensions) in the middle compared to Input and Output layers.

Source: https://towardsdatascience.com/autoencoders-ae-a-smart-way-to-process-your-data-using-unsupervised-neural-networks-9661f93a8509

# Neural network

## Autoencoders (AE)




Source: https://towardsdatascience.com/autoencoders-ae-a-smart-way-to-process-your-data-using-unsupervised-neural-networks-9661f93a8509

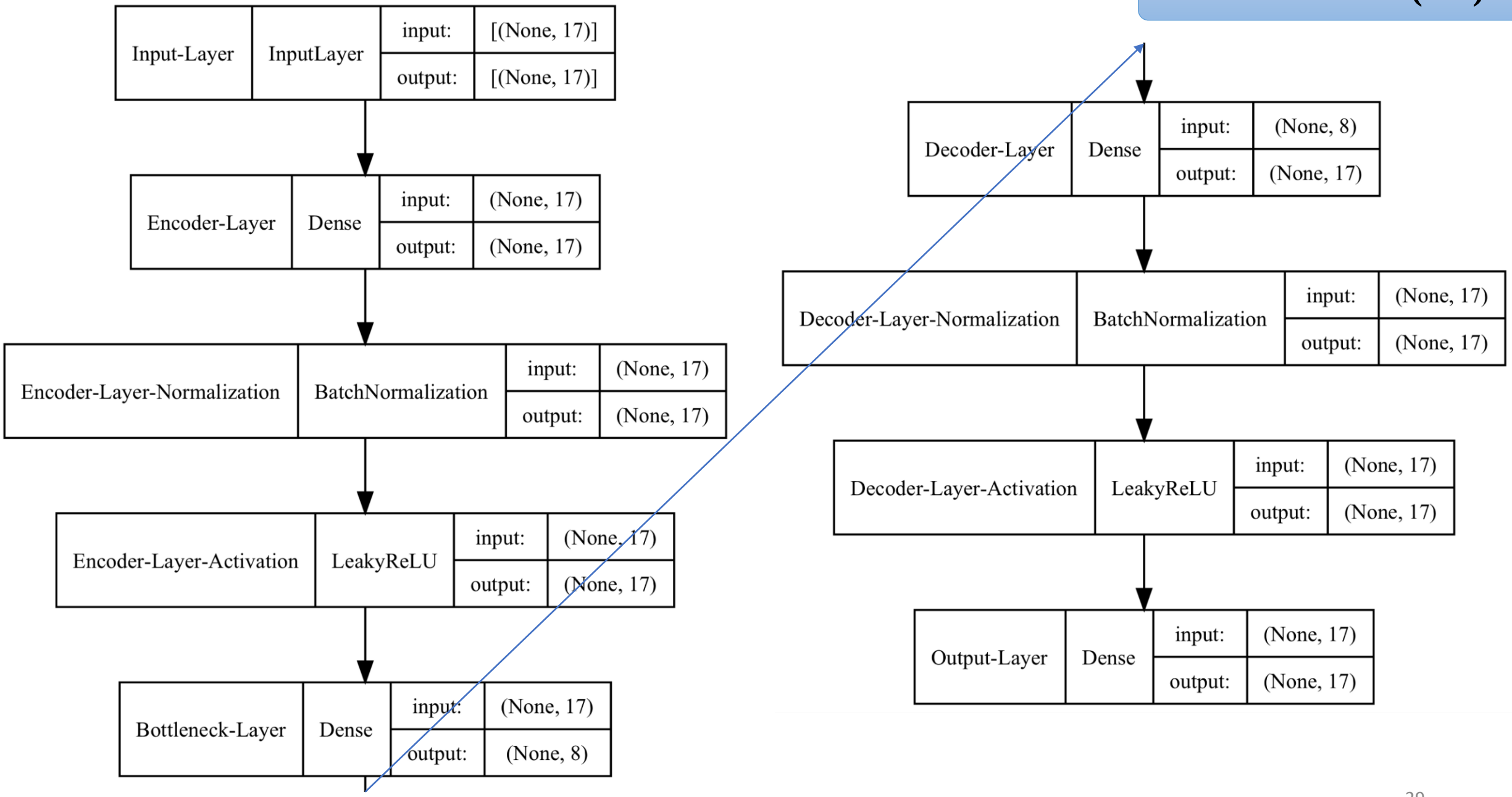
# Neural network

## Autoencoders (AE)

- Autoencoders have Input, Hidden and Output layers similar to that of other types of Neural Networks.
- Hidden layers of Autoencoders contain two significant parts: Encoder and Decoder.
- Output nodes within an Autoencoder match the input nodes.
  - Hence, the Autoencoder Neural Network tries to recreate the same feature values that it receives in the Input layer.
  - Since we are trying to recreate (predict) features themselves, we do not require labelled target data.
  - Hence, we can refer to Autoencoders as Unsupervised models, although some literature refers to them as Self-Supervised models.

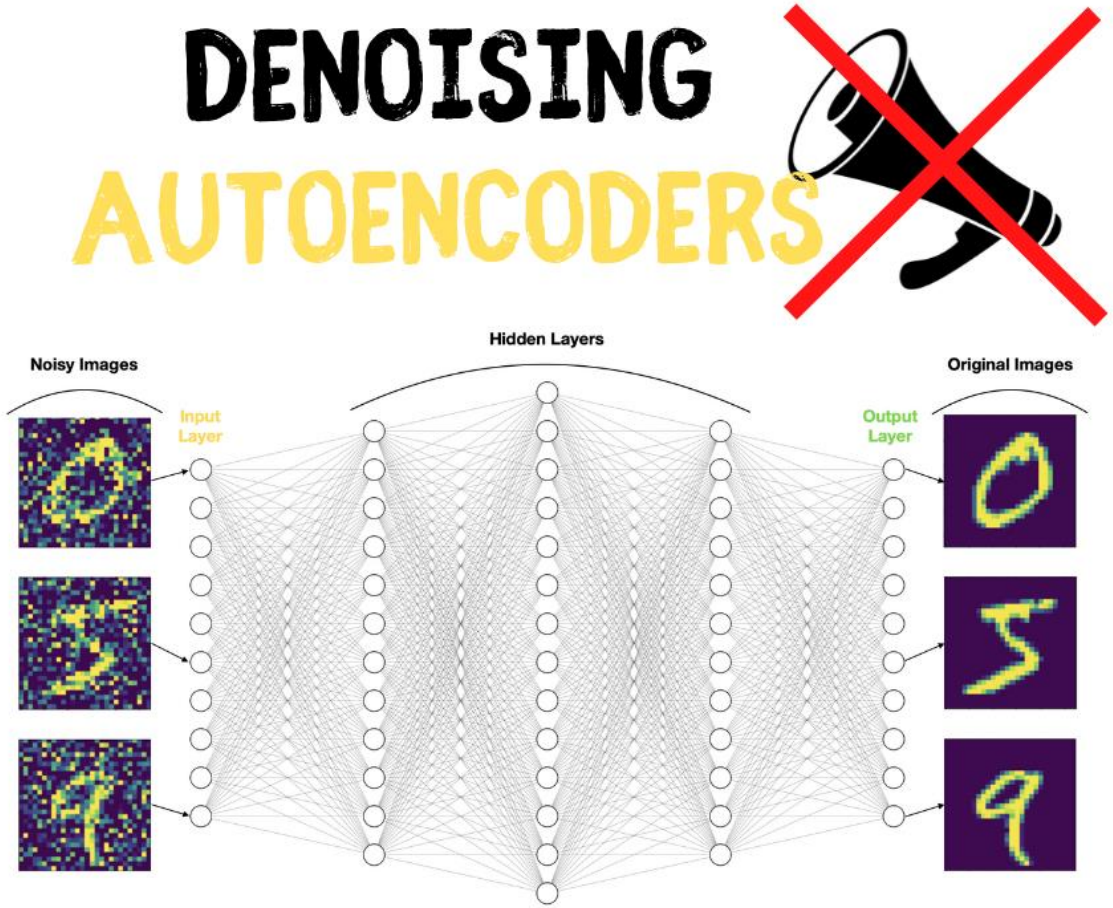
Source: https://towardsdatascience.com/autoencoders-ae-a-smart-way-to-process-your-data-using-unsupervised-neural-networks-9661f93a8509

# Autoencoders (AE)



# Neural network

## Denoising Autoencoders (DAE)




Source: https://towardsdatascience.com/denoising-autoencoders-dae-how-to-use-neural-networks-to-clean-up-your-data-cd9c19bc6915

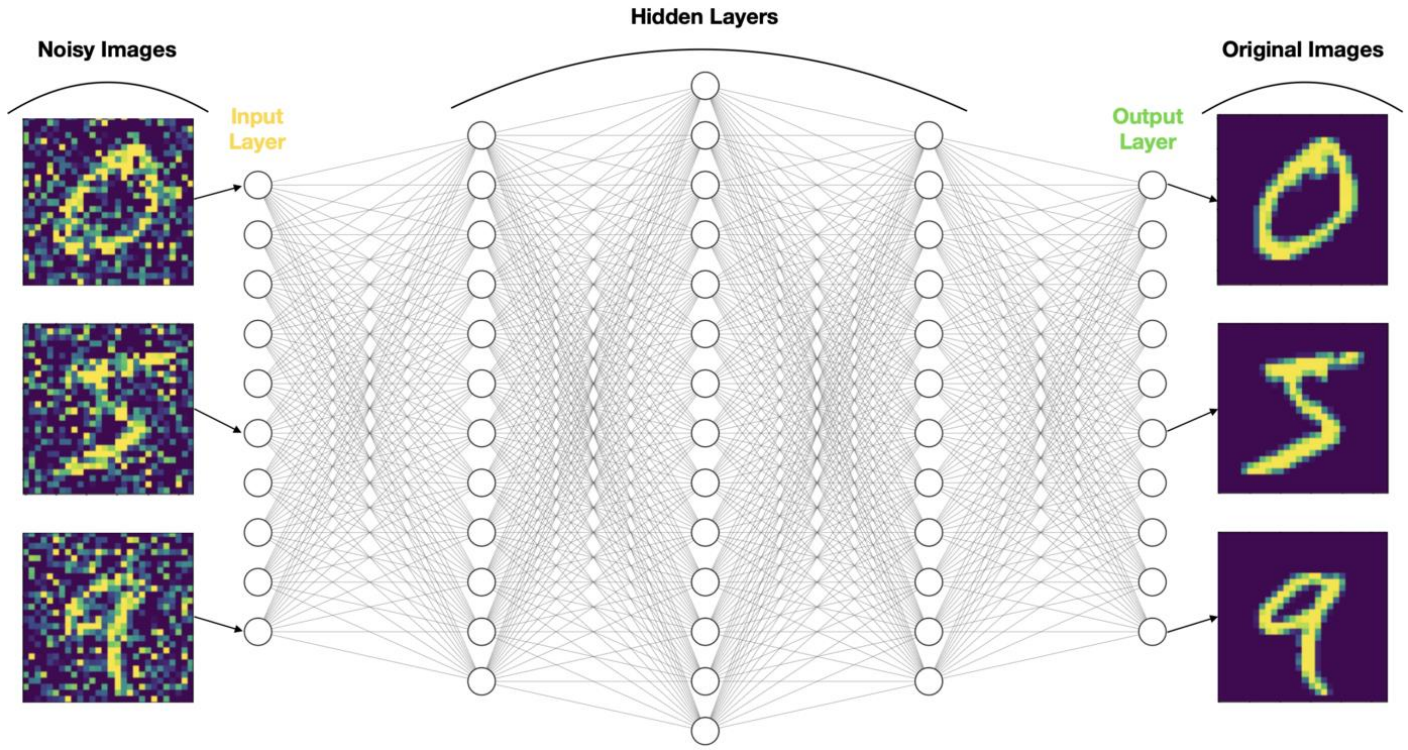
## Denoising Autoencoders (DAE)

- The purpose of a DAE is to remove noise.
- Different from other popular types of Neural Networks (Feed-Forward, Recurrent and Convolutional) because they do not require labelled data to train them.
  - Hence, we can refer to them as Unsupervised or, if we want to be very precise, as Self-Supervised Neural Networks.
- To achieve this equilibrium of matching target outputs to inputs, denoising autoencoders accomplish this goal in a specific way.
  - The program takes in a corrupted version of some model.
  - Try to reconstruct a clean model through the use of denoising techniques.
  - Apply noise in a particular amount as a percentage of the model.
  - Try to force the hidden layer to work from the corrupted version to produce a clean version.
- Denoising autoencoders can also be stacked on each other to provide iterative learning toward this key goal.

Source: https://towardsdatascience.com/denoising-autoencoders-dae-how-to-use-neural-networks-to-clean-up-your-data-cd9c19bc6915
https://www.techopedia.com/definition/33286/denoising-autoencoder-dae

# Neural network

## Denoising Autoencoders (DAE)



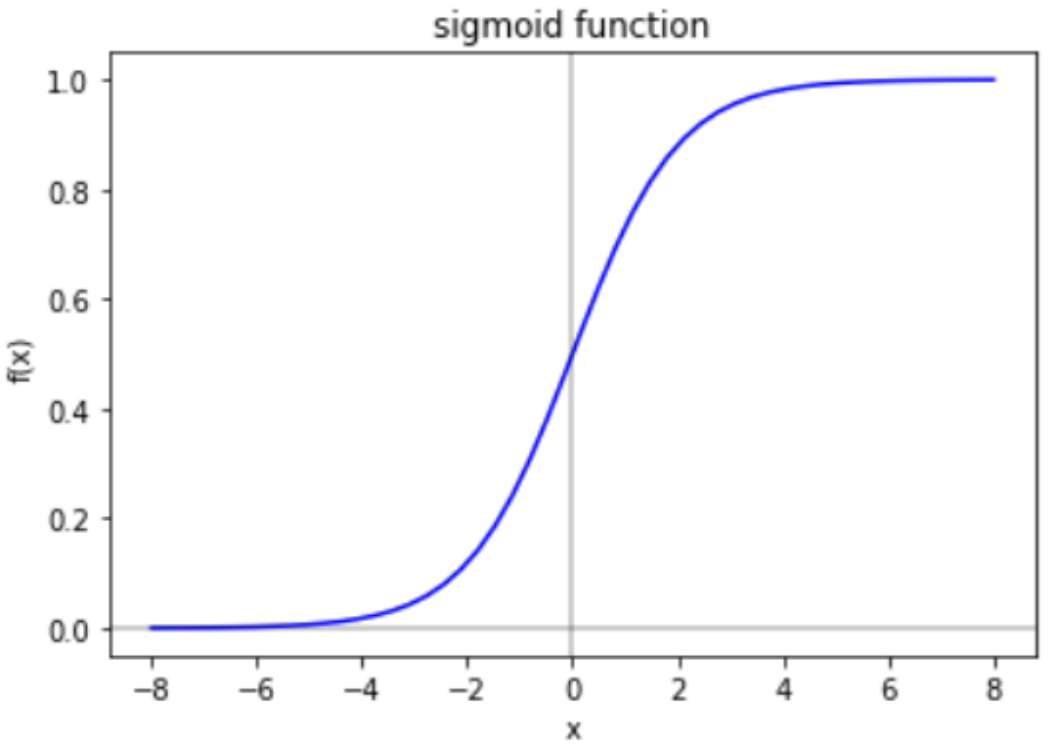

- Unlike Undercomplete AE, we may use the same or higher number of neurons within the hidden layer, making the DAE overcomplete.
- The outputs are the original data (e.g., images), while the inputs contain data with some added noise.

Source: https://towardsdatascience.com/denoising-autoencoders-dae-how-to-use-neural-networks-to-clean-up-your-data-cd9c19bc6915

# Neural network

## Activation function

*sigmoid*




- Properties
  - $f(x) \in [0, 1]$
  - $x \in [-\infty, \infty]$
  - $x \rightarrow 0$, f(x) becomes linear
  - abs(x) > 4, f(x) changes slowly

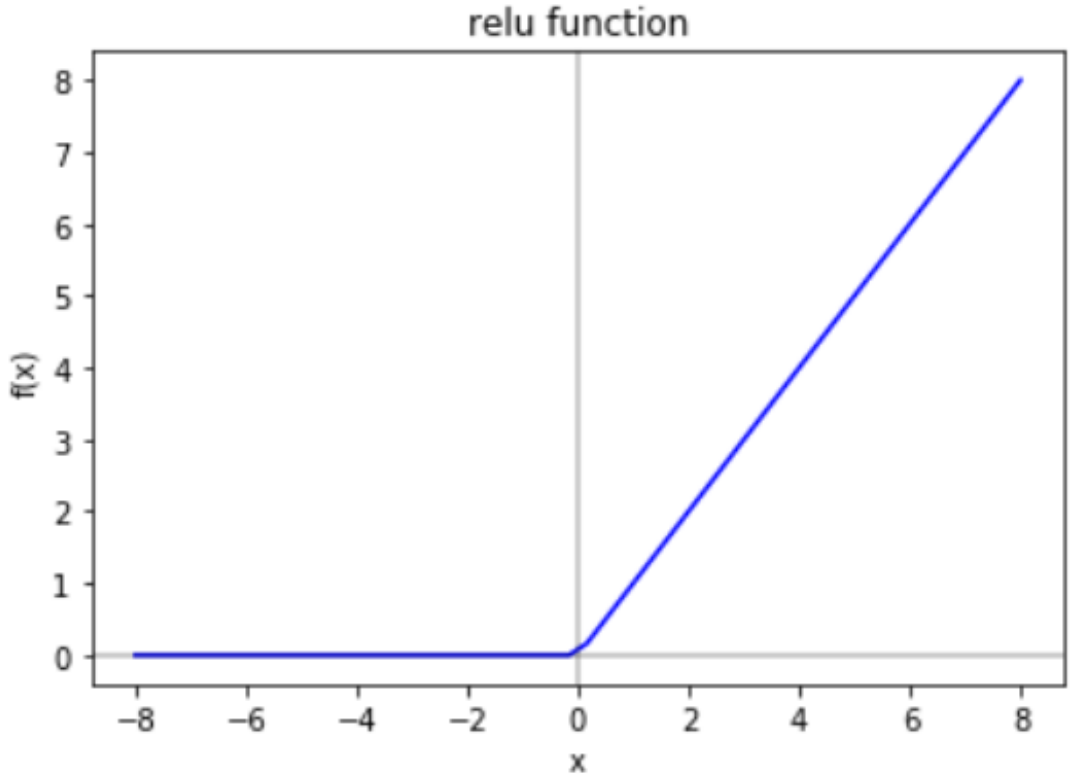


Source: RADS 602: Data Mining and Machine Learning, Ratchainant Thammasudjarit (Ph.D.)

# Neural network

## Activation function

*ReLu (Rectifier Linear Unit)*




- Properties
  - $f(x) \in [0, \infty]$
  - $x \in [-\infty, \infty]$
  - $x \leq 0, f(x) = 0$
  - $x > 0, f(x) = x$

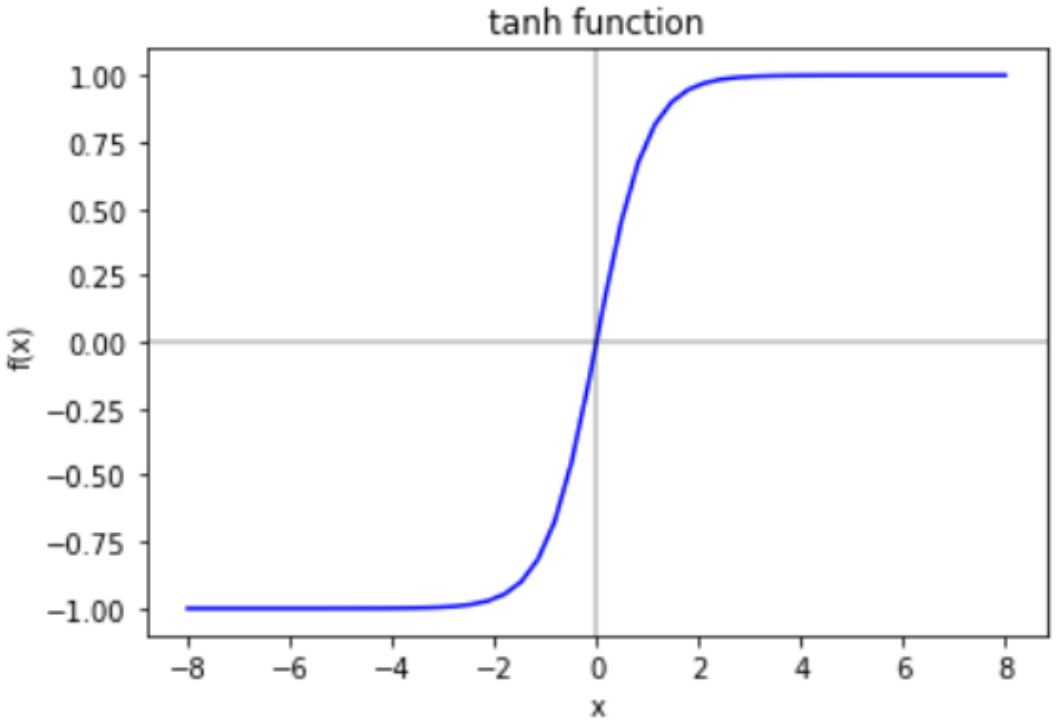


Source: RADS 602: Data Mining and Machine Learning, Ratchainant Thammasudjarit (Ph.D.)

# Neural network

## Activation function

*tanh*



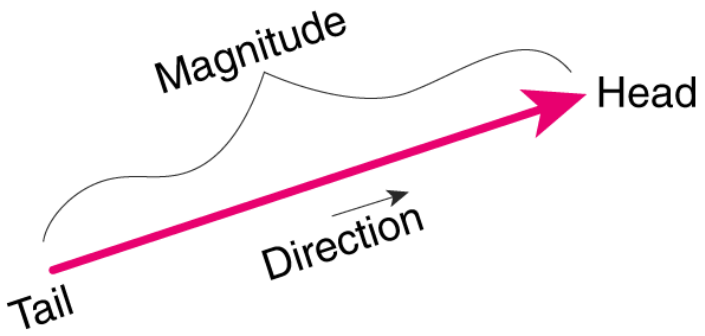

- Properties
  - f(x) ∈ [-1, 1]
  - x ∈ [-∞, ∞]
  - x ⟶ 0, f(x) becomes linear
  - abs(x) > 2, f(x) changes slowly

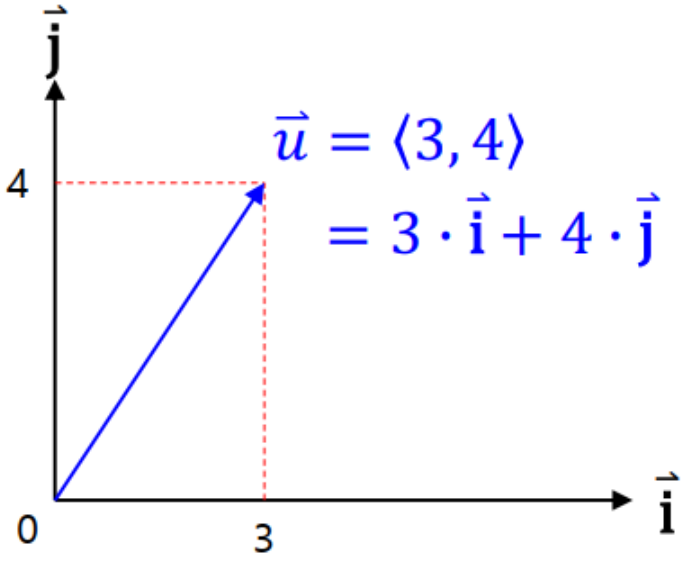


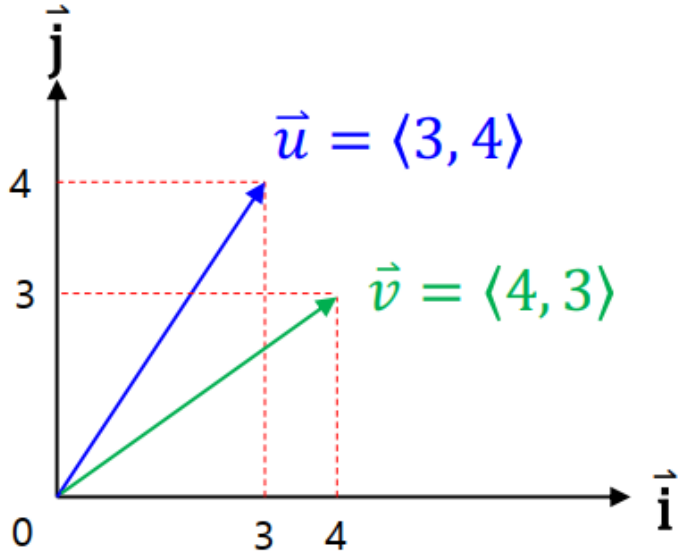
Source: RADS 602: Data Mining and Machine Learning, Ratchainant Thammasudjarit (Ph.D.)

# Neural network

## Vector




$$\vec{u} \cdot \vec{v} = (3 \times 4) + (4 \times 3) = 24$$

$$\|\vec{u}\| = \sqrt{3^2 + 4^2} = 5$$



Source: RADS 602: Data Mining and Machine Learning, Ratchainant Thammasudjarit (Ph.D.)

# Neural network

## Matrix

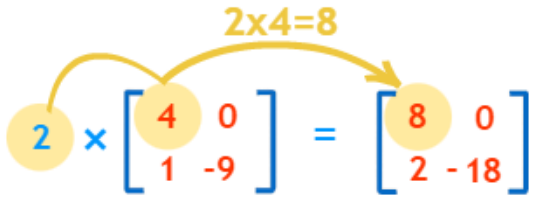
- **A stack of vectors, an array of numbers.**

$$\begin{bmatrix} 6 & 4 & 24 \\ 1 & -9 & 8 \end{bmatrix}$$

A Matrix

(This one has 2 Rows and 3 Columns)

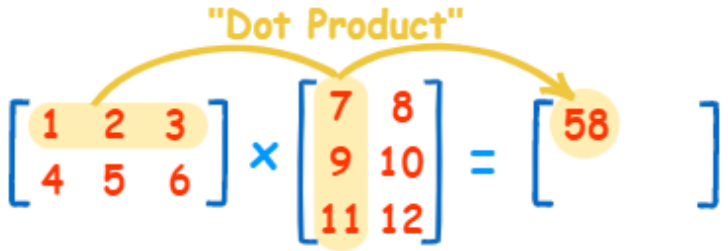
### Scalar multiplication



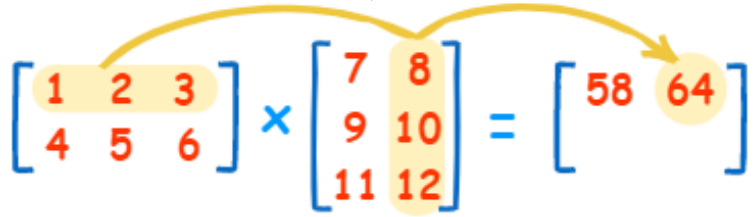

These are the calculations:

| | |
|---|---|
| 2×4=8 | 2×0=0 |
| 2×1=2 | 2×-9=-18 |

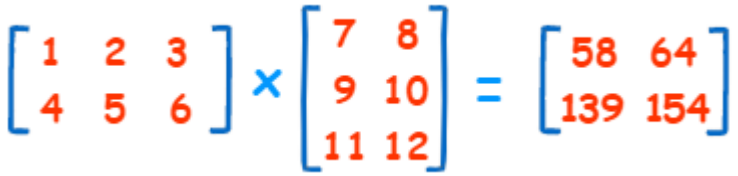
### Multiplying a matrix by another matrix (Dot Product)




$(1, 2, 3) \bullet (7, 9, 11) = 1\times7 + 2\times9 + 3\times11 = 58$




$(1, 2, 3) \bullet (8, 10, 12) = 1\times8 + 2\times10 + 3\times12 = 64$




Source: https://www.mathsisfun.com/algebra/matrix-multiplying.html

# Neural network

## Performance evaluation metrics

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

$$specificity = \frac{TN}{TN + FP}$$

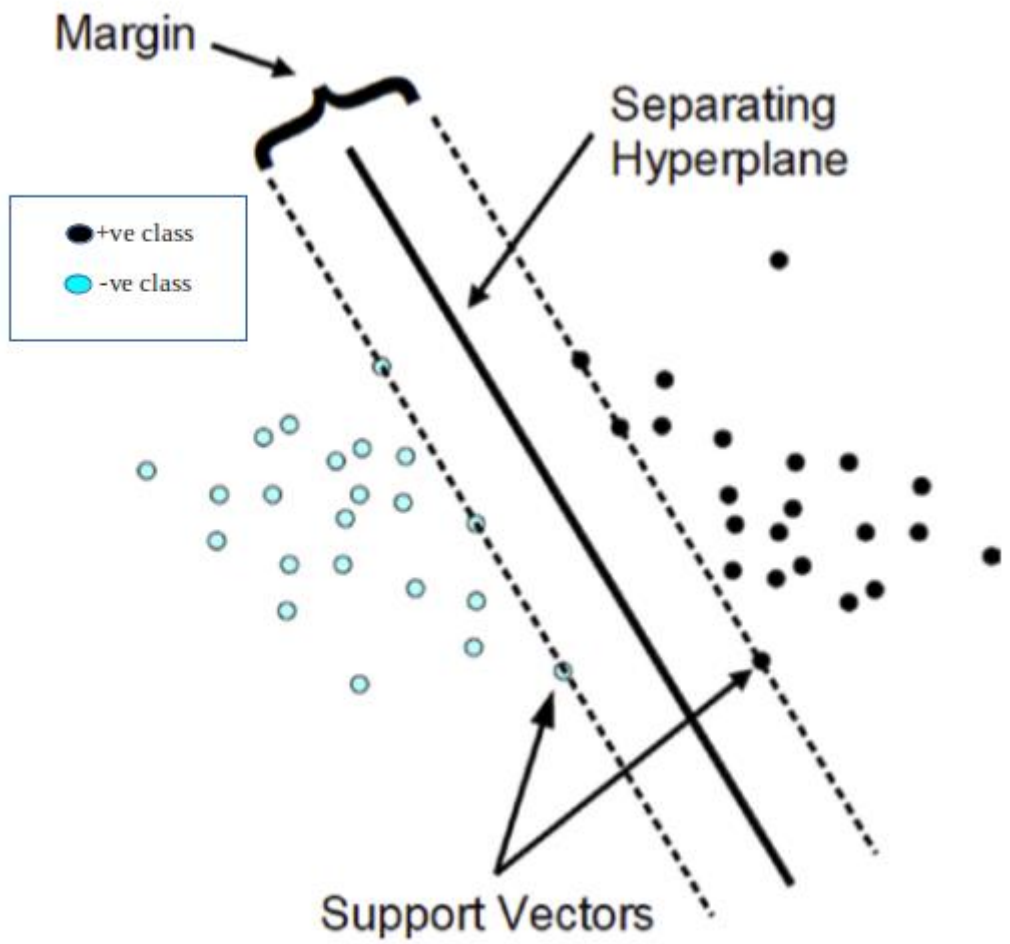
$$F1 = \frac{2PR}{P + R}$$



Source: RADS 602: Data Mining and Machine Learning, Ratchainant Thammasudjarit (Ph.D.)

# Support vector machines

## Introduction to SVM

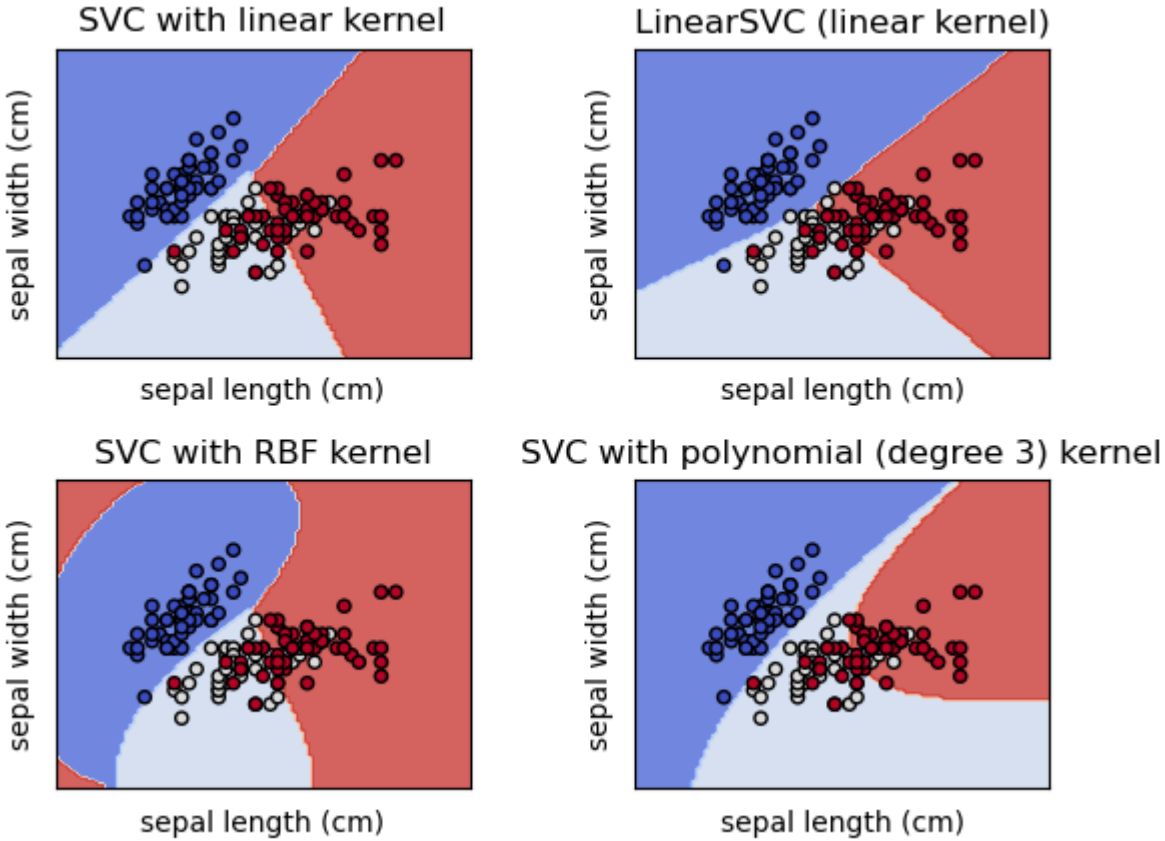



- SVM is a supervised learning algorithm.
- Can be used for both classification and regression problems.
- Mostly used for classification problems.

Source: https://en.wikipedia.org/wiki/Support-vector_machine
https://scikit-learn.org/stable/modules/svm.html#:~:text=Support%20vector%20machines%20(SVMs)%20are,than%20the%20number%20of%20samples.

# Support vector machines

## Kernels




- More kernels available. Linear, RBF, Polynomial.
- Support binary or more classes.

☑ Kernel = Linear

Source: https://en.wikipedia.org/wiki/Support-vector_machine
https://scikit-learn.org/stable/modules/svm.html#:~:text=Support%20vector%20machines%20(SVMs)%20are,than%20the%20number%20of%20samples.

# Support vector machines

## Hinge Loss Formula

- The loss is defined according to the following formula:

$$l(y) = max(0, 1 - t \cdot y)$$

Where:

- t is the actual outcome (either 1 or -1)
- y is the output of the classifier

- **Example**: the outcome was **1**, and the prediction was 0.5.

$$l(y) = max(0, 1 - 1 \cdot 0.5) = 0.5$$

- **Example**: the outcome was **-1**, and the prediction was 0.5.

$$l(y) = max(0, 1 - (-1) \cdot 0.5) = 1.5$$

Squared hinge = (Hinge loss)$^2$



Source: https://programmathically.com/understanding-hinge-loss-and-the-svm-cost-function/

# Support vector machines

## Penalty

**L1 regularization: Lasso Regression**

**(Least Absolute Shrinkage and Selection Operator)**

- Suitable for assumption: good prediction come from less predictors.
- Shrinks the less important feature's coefficient to zero.
  - ➢ Thus, removing some feature altogether.
  - ➢ Reduce variance, prevent overfitting.
- Works well for feature selection in case we have a huge number of features.

**L2 regularization: Ridge Regression**

- Suitable for assumption: good prediction come from many predictors.
- Shrinks the feature's coefficient to small but not zero.
  - ➢ Reduce variance, prevent overfitting.

☑ Penalty = L2



Source: https://towardsdatascience.com/l1-and-l2-regularization-methods-ce25e7fc831c

# Support vector machines

## Class Weight

- Set the parameter C of class i to class_weight[i]*C for SVC.
- If not given, all classes are supposed to have weight one.
- The “balanced” mode uses the values of y to automatically adjust weights inversely proportional to class frequencies in the input data as
  - n_samples / (n_classes * np.bincount(y)).

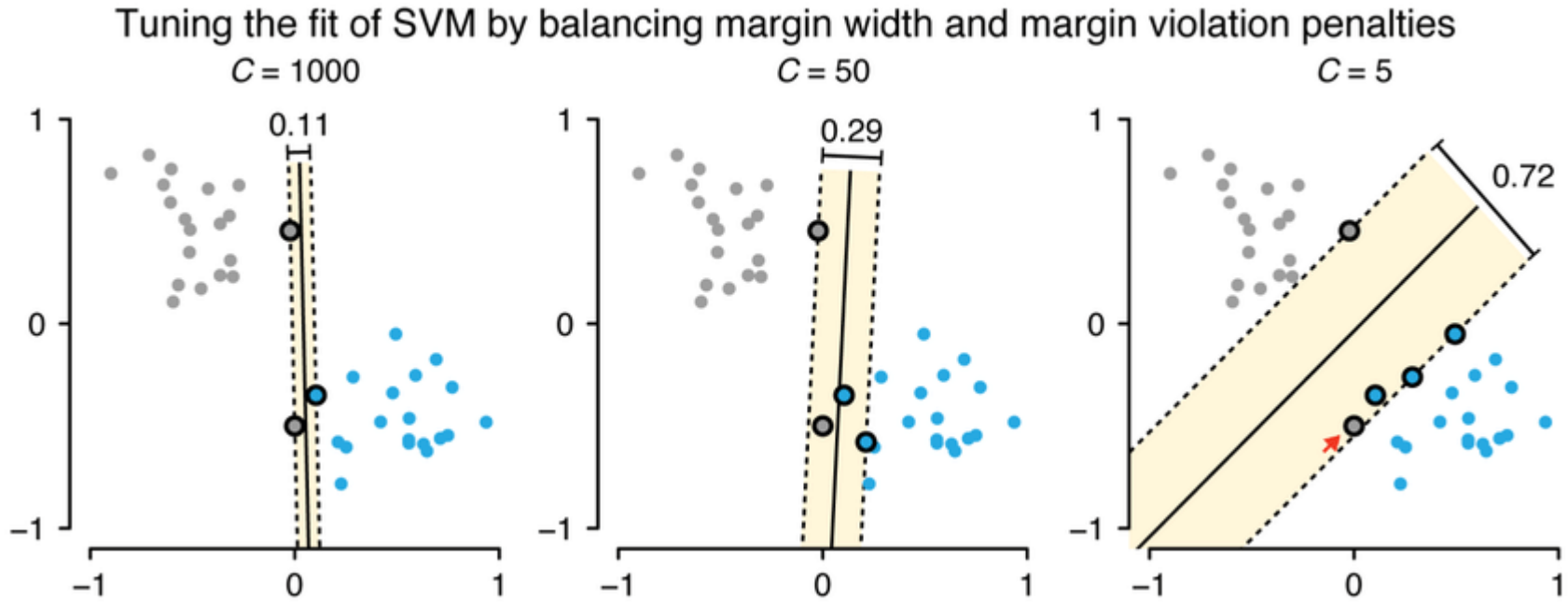
☑ Class weight = Balanced



Source: https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html

# Support vector machines

## Regularization parameter (C)




- Must be strictly positive.
- The strength of the regularization is inversely proportional to C.
- If the values of C are very small the margin increases thus Soft SVM.
- Large value of C can cause overfitting therefore we need to select the correct value using Hyperparameter Tuning.

$C=0.000001\ (e^{-06})$

Source: https://www.analyticsvidhya.com/blog/2021/04/insight-into-svm-support-vector-machine-along-with-code/#:~:text=%E2%80%93%20If%20the%20values%20of%20C,correct%20value%20using%20Hyperparameter%20Tuning.

# Support vector machines

## Advantages and disadvantages

**The advantages:**

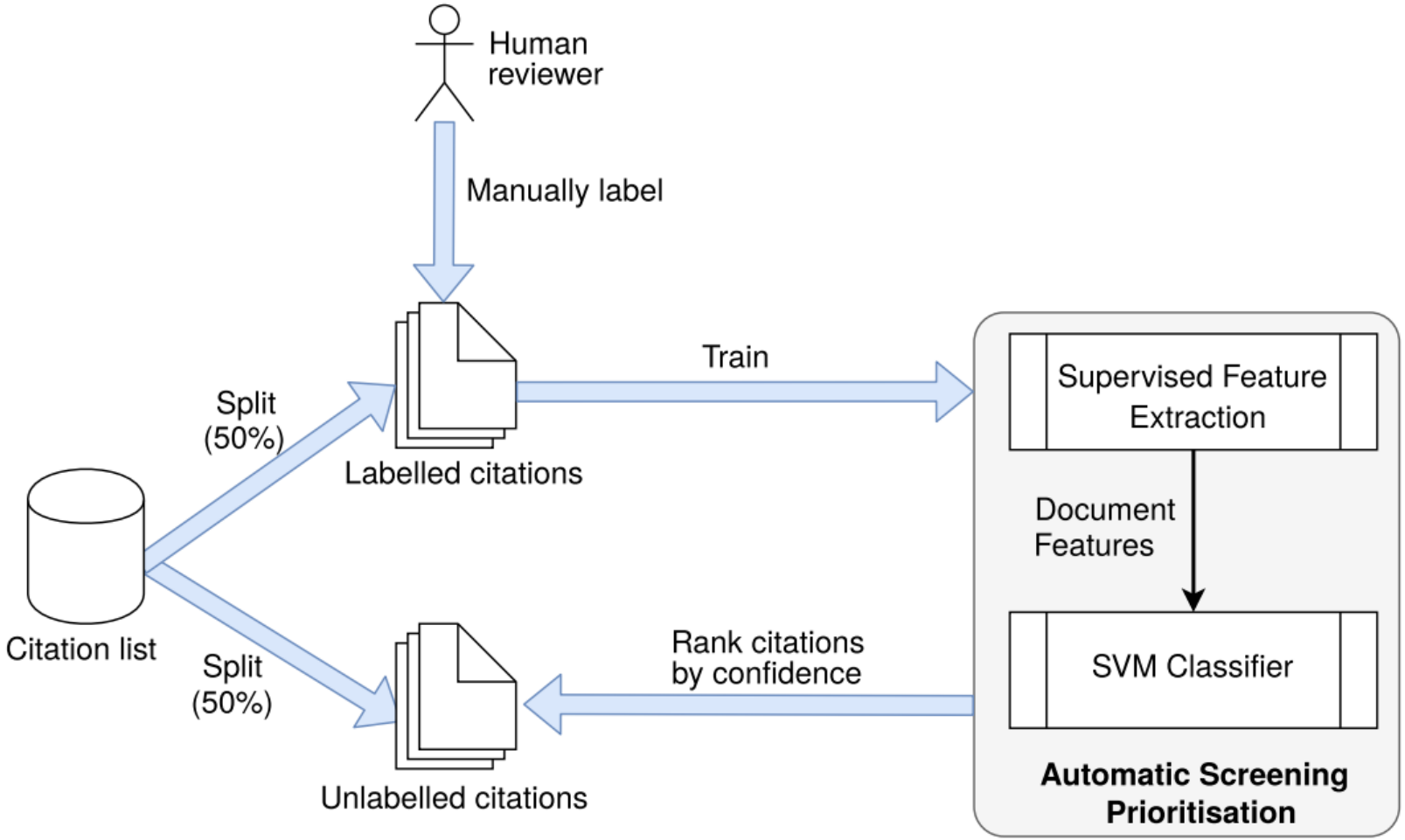
- Effective in high dimensional spaces.
- Still effective in cases where number of dimensions is greater than the number of samples.
- Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
- Versatile: different Kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

**The disadvantages:**

- If the number of features is much greater than the number of samples, avoid over-fitting in choosing Kernel functions and regularization term is crucial.
- SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation.
- Does not execute well when the data set has more noise i.e. target classes are overlapping.
- Long training time for large datasets (not suitable for large datasets).

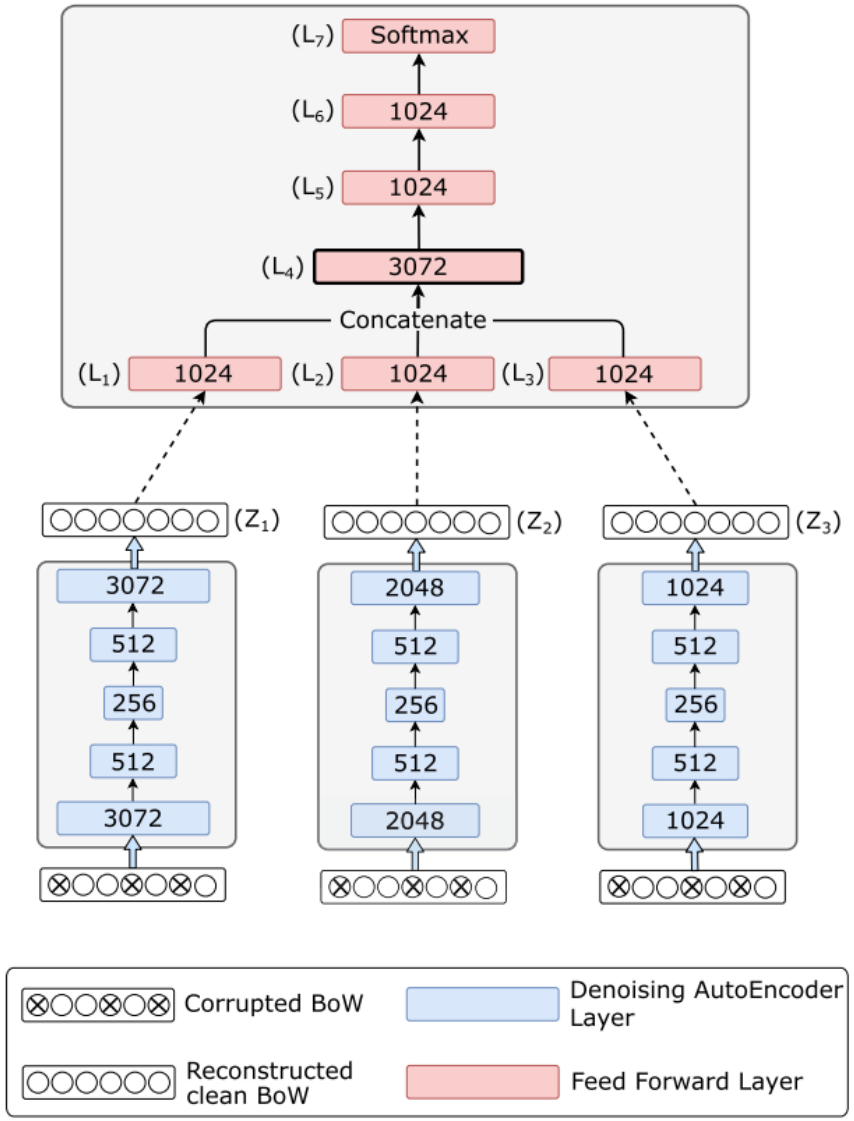
# Methodology

## Automatic screening prioritisation framework



# Methodology

## Supervised feature extraction




1. The proposed method coordinates two types of neural networks:
   - a denoising autoencoder and
   - a feed forward network
2. A denoising autoencoder aims to reconstruct the input BoW feature space given an artificially corrupted version of the BoW space.
3. We artificially corrupt the input BoW feature using additive Gaussian noise.
4. The goal of an one-layer denoising autoencoder is to firstly encode the corrupted feature vector into a lower dimensional representation using the encoder mapping function.
5. The encoded representation is then mapped back, i.e. decoded, into a BoW reconstruction through the decoder mapping function.

# Experiments

## Data

**Table 1**
23 publicly available review datasets used in the experiments of this paper.

| Source | Dataset | # citations | (%) eligible citations | Bibliographic metadata |
|---|---|---|---|---|
| Clinical (Wallace et al., 2010) | COPD | 1606 | 12.2 | ✗ |
| | Proton Beam | 4751 | 5.1 | ✗ |
| | Micro Nutrients | 4010 | 6.4 | ✗ |
| | ACEInhibitors | 2544 | 1.6 | ✓ |
| | ADHD | 851 | 2.4 | ✓ |
| | Antihistamines | 310 | 5.2 | ✓ |
| | Atypical Antipsychotics | 1120 | 13.0 | ✓ |
| | Beta Blockers | 2072 | 2.0 | ✓ |
| | Calcium Channel Blockers | 1218 | 8.2 | ✓ |
| | Estrogens | 368 | 21.7 | ✓ |
| Drug (Cohen et al., 2006) | NSAIDs | 393 | 10.4 | ✓ |
| | Opioids | 1915 | 0.8 | ✓ |
| | Oral Hypoglycemics | 503 | 27.0 | ✓ |
| | Proton PumpInhibitors | 1333 | 3.8 | ✓ |
| | Skeletal Muscle Relaxants | 1643 | 0.5 | ✓ |
| | Statins | 3465 | 2.5 | ✓ |
| | Triptans | 671 | 3.6 | ✓ |
| | Urinary Incontinence | 327 | 12.2 | ✓ |
| | PFOA/PFOS | 6330 | 1.5 | ✓ |
| SWIFT (Howard et al., 2016) | Bisphenol A (BPA) | 7699 | 1.4 | ✓ |
| | Transgenerational | 48,637 | 1.6 | ✓ |
| | Fluoride and neurotoxicity | 4479 | 1.1 | ✗ |
| | Neuropathic pain | 29,207 | 17.2 | ✗ |

# Experiments

## Evaluation settings

- Work Saved over Sampling at r% recall (WSS@r%).
- Report average values of the WSS@95% and precision@95%recall metrics over 10 cross-validation folds.

$$WSS@r\% = \underbrace{\frac{TN+FN}{N}}_{\text{(\%) excluded citations}} - \overbrace{(1-r)}^{\text{penalty term}} \tag{5}$$

$$WSS@95\% = \frac{TN+FN}{N} - (1-0.95) = \frac{TN+FN}{N} - 0.05 \tag{6}$$

# Experiments

## Automatic prioritisation system

1. Employs an L2-regularised linear SVM classifier to rank the citations according to the signed-margin distance between the citation feature vectors and the SVM hyperplane.
2. In order to better account for the class imbalance between eligible and ineligible citations,
   - Used a reduced misclassification cost, i.e. a trade-off between maximising the margin between the two classes and minimising classification errors,
     - by setting the regularisation parameter $C = 1 \times 10^{-6}$.
   - We used the same hyper-parameter settings for the SVM classifier across all review datasets and across all feature extraction methods.

# Experiments

## Baseline methods

- **Baseline feature extraction methods with selected parameter settings used in the experiments of this paper**

**Table 2**
Baseline feature extraction methods with selected parameter settings used in the experiments of this paper.

| Baseline method | Hyper-parameters |
|---|---|
| BoW | Top (Stemmed) words: 10,000 |
| SVD | eigenvalues: 300 |
| LDA | topics: 300, iterations: 500 |
| PV | topics: 300, iterations: 500, document vector: 1000, word vector: 300 |
| MeSH tags (bibliographic metadata) | – |

# Experiments

## Result: *Hyper-parameter settings*

- **Hyperparameter settings of the supervised feature extraction method**

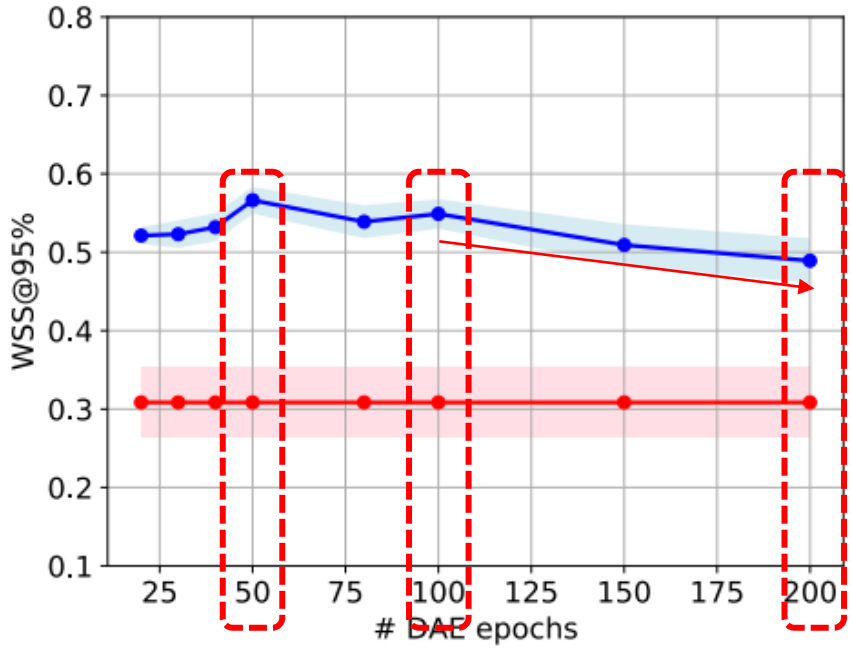
**Table 3**
Hyperparameter settings of the supervised feature extraction method.

| Hyper-parameter | Value |
|---|---|
| size of minibatch (DAE) | 32 |
| dropout regularisation | 0.7 |
| number of training epochs (FF) | 100 |
| size of minibatch (FF) | 128 |

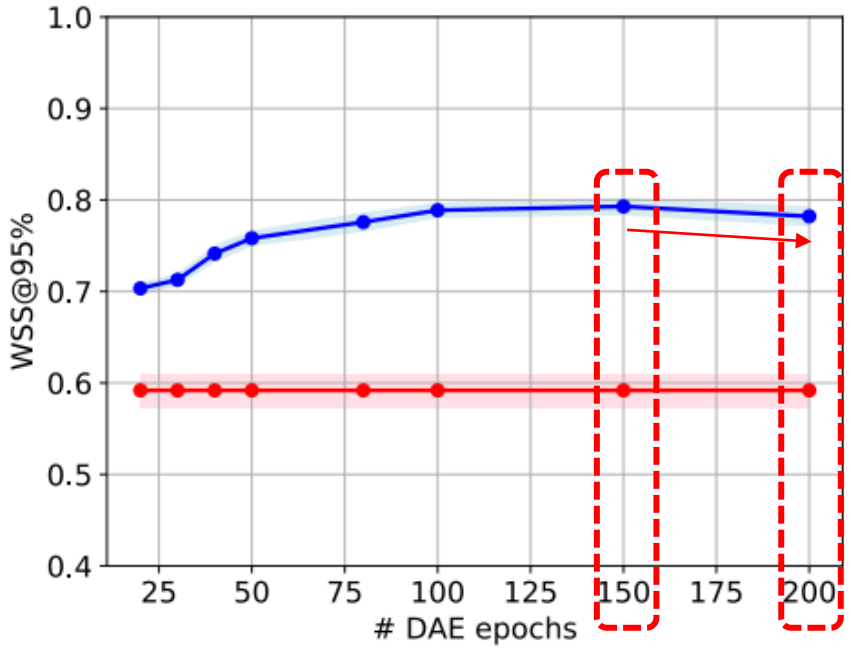
# Experiments

## Result: *Effect of number of DAE epochs*

- **WSS@95% performance of the proposed method (i.e. DAE-FF) on an increasing number of DAE epochs across the Statins and BPA development reviews**




(a) $WSS@95\%$ performance on the Statins development review




(b) $WSS@95\%$ performance on the BPA development review

**Fig. 3.** *WSS*@95% performance of the proposed method (i.e. DAE-FF) on an increasing number of DAE epochs across the Statins and BPA development reviews. The figures also illustrate the *WSS*@95% performance of the BoW baseline method. The thick lines are average *WSS*@95% values. The bands surrounding the thick lines represent the 95% confidence interval of the mean *WSS*@95% values across 10 validation rounds.

# Experiments

## Result: *Effect of model architecture*

- **The WSS@95% performance of the 7 model architectures on the Statins and BPA development reviews**

**Table 4**

*WSS*@95% performance of 7 different network architectures of our method (i.e. *model*_1 to *model*_7) on the two development reviews. The superscript ** shows that the corresponding model obtained a statistically significant lower performance when compared to the *WSS*@95% performance of *model*_7 according to a two-tailed paired *t*-test at $p < 0.01$ level. The superscript * denotes statistically significant difference at $p < 0.05$ level.

| Model | *DAE*_1 | *DAE*_2 | *DAE*_3 | WSS@95% | |
|---|---|---|---|---|---|
| | (1024,512, 256,512,1024) | (2048,512, 256,512,2048) | (3072,512, 256,512,3072) | Statins | BPA |
| *model*_1 | – | – | – | 0.414** | 0.687** |
| *model*_2 | √ | – | – | 0.514** | 0.709** |
| *model*_3 | – | √ | – | 0.488** | 0.697** |
| *model*_4 | – | – | √ | 0.492** | 0.703** |
| *model*_5 | √ | √ | – | 0.534 | 0.786 |
| *model*_6 | √ | – | √ | 0.555 | 0.773* |
| *model*_7 | √ | √ | √ | **0.566** | **0.792** |

# Experiments

## Result: *Comparison with baseline methods*

- **WSS@95% performance of our method against 5 single-view feature extraction baselines**

**Table 5**

*WSS@95%* performance of our method against 5 single-view feature extraction baselines. *WSS@95%* scores are averages across 10 validation runs for each of the 23 review datasets. The superscript ** shows that the DAE-FF method achieved a statistically significant better performance according to a two-tailed paired *t*-test over all 5 baseline methods at $p < 0.01$ level. The superscript * denotes statistically significant improvements over the 5 baselines at $p < 0.05$ level.

| Dataset | BoW | SVD | LDA | PV | MeSH | DAE-FF |
|---|---|---|---|---|---|---|
| COPD | 0.458 | 0.605 | 0.555 | 0.633 | – | **0.666** |
| Proton Beam | 0.746 | 0.722 | 0.787 | 0.709 | – | **0.816**** |
| Micro Nutrients | 0.510 | 0.597 | 0.430 | 0.590 | – | **0.662*** |
| ACEInhibitors | 0.752 | **0.791** | 0.548 | 0.708 | 0.375 | 0.787 |
| ADHD | **0.744** | 0.712 | 0.485 | 0.481 | 0.567 | 0.665 |
| Antihistamines | 0.048 | 0.053 | 0.042 | 0.211 | 0.192 | **0.310** |
| Atypical Antipsychotics | 0.136 | 0.038 | 0.076 | 0.150 | 0.199 | **0.329**** |
| Beta Blockers | 0.470 | 0.455 | 0.507 | 0.130 | 0.237 | **0.587** |
| Calcium Channel Blockers | 0.177 | 0.262 | 0.234 | 0.169 | 0.130 | **0.424**** |
| Estrogens | 0.288 | 0.292 | 0.360 | 0.271 | 0.238 | **0.397** |
| NSAIDs | 0.719 | 0.698 | 0.569 | 0.593 | 0.331 | **0.723** |
| Opioids | 0.304 | 0.251 | 0.350 | 0.472 | 0.116 | **0.533** |
| Oral Hypoglycemics | 0.081 | 0.046 | **0.106** | 0.055 | 0.065 | 0.095 |
| Proton PumpInhibitors | 0.239 | 0.299 | 0.293 | **0.503** | 0.323 | 0.400 |
| Skeletal Muscle Relaxants | 0.102 | 0.186 | 0.148 | **0.345** | 0.050 | 0.286 |
| Statins | 0.309 | 0.306 | 0.415 | 0.293 | 0.236 | **0.566**** |
| Triptans | **0.417** | 0.356 | 0.331 | 0.295 | 0.241 | 0.310 |
| Urinary Incontinence | 0.291 | 0.504 | 0.443 | 0.451 | 0.220 | **0.531** |
| PFOA/PFOS | 0.773 | 0.794 | 0.797 | 0.833 | 0.405 | **0.848**** |
| Bisphenol A (BPA) | 0.591 | 0.709 | 0.702 | 0.629 | 0.631 | **0.793**** |
| Transgenerational | 0.619 | 0.579 | 0.612 | 0.542 | 0.432 | **0.707**** |
| Fluoride and neurotoxicity | 0.719 | 0.843 | **0.847** | 0.828 | – | 0.799 |
| Neuropathic pain | 0.471 | 0.428 | 0.534 | 0.442 | – | **0.608**** |
| Average (all datasets) | 0.433 | 0.458 | 0.442 | 0.449 | 0.277 | **0.564** |

# Experiments

## Result: *Comparison with baseline methods*

- **Compares the performance of the DAE-FF method against 5 composite feature extraction methods.**

**Table 6**
*WSS@95%* performance of our method against 5 composite feature extraction methods (i.e. column-wide concatenation of different single-view feature spaces).

| Dataset | BoW-SVD | BoW-LDA | BoW-PV | BoW-MeSH | BoW-SVD-LDA-PV | DAE-FF |
|---|---|---|---|---|---|---|
| COPD | 0.598 | 0.609 | 0.599 | — | 0.640 | **0.666** |
| Proton Beam | 0.734 | 0.778 | 0.733 | — | 0.772 | **0.816**** |
| Micro Nutrients | 0.568 | 0.416 | 0.574 | — | 0.607 | **0.662**** |
| ACEInhibitors | 0.798 | **0.801** | 0.798 | 0.773 | 0.768 | 0.787 |
| ADHD | 0.719 | 0.624 | 0.719 | **0.738** | 0.633 | 0.665 |
| Antihistamines | 0.053 | 0.229 | 0.054 | 0.273 | 0.253 | **0.310** |
| Atypical Antipsychotics | 0.042 | 0.152 | 0.040 | 0.134 | 0.148 | **0.329**** |
| Beta Blockers | 0.469 | 0.532 | 0.468 | 0.552 | 0.499 | **0.587** |
| Calcium Channel Blockers | 0.249 | 0.308 | 0.250 | 0.398 | 0.291 | **0.424** |
| Estrogens | 0.297 | 0.300 | 0.295 | **0.408** | 0.293 | 0.397 |
| NSAIDs | 0.699 | 0.684 | 0.698 | 0.595 | 0.692 | **0.723** |
| Opioids | 0.256 | 0.318 | 0.255 | 0.332 | 0.296 | **0.533** |
| Oral Hypoglycemics | 0.042 | **.114** | 0.043 | 0.112 | 0.109 | 0.095 |
| Proton PumpInhibitors | 0.304 | 0.302 | 0.305 | 0.252 | 0.345 | **0.400** |
| Skeletal Muscle Relaxants | 0.182 | **0.465** | 0.184 | 0.318 | 0.435 | 0.286 |
| Statins | 0.316 | 0.364 | 0.311 | 0.252 | 0.398 | **0.566**** |
| Triptans | 0.366 | 0.437 | 0.361 | 0.241 | **0.445** | 0.434 |
| Urinary Incontinence | 0.500 | 0.381 | 0.504 | 0.426 | 0.362 | **0.531** |
| PFOA/PFOS | 0.819 | 0.833 | 0.796 | 0.815 | 0.826 | **0.848**** |
| Bisphenol A (BPA) | 0.759 | **0.775** | 0.690 | 0.717 | 0.711 | 0.758 |
| Transgenerational | 0.598 | 0.646 | 0.576 | 0.641 | 0.644 | **0.707**** |
| Fluoride and neurotoxicity | 0.835 | 0.778 | 0.835 | — | **0.849** | 0.799 |
| Neuropathic pain | 0.484 | 0.472 | 0.441 | — | 0.477 | **0.608**** |
| Average (all datasets) | 0.465 | 0.492 | 0.458 | 0.450 | 0.500 | **0.564** |

# Experiments

## Result: *Comparison with baseline methods*

- **The average precision at recall level of 95% obtained by our proposed DAE-FF across the 23 review datasets.**

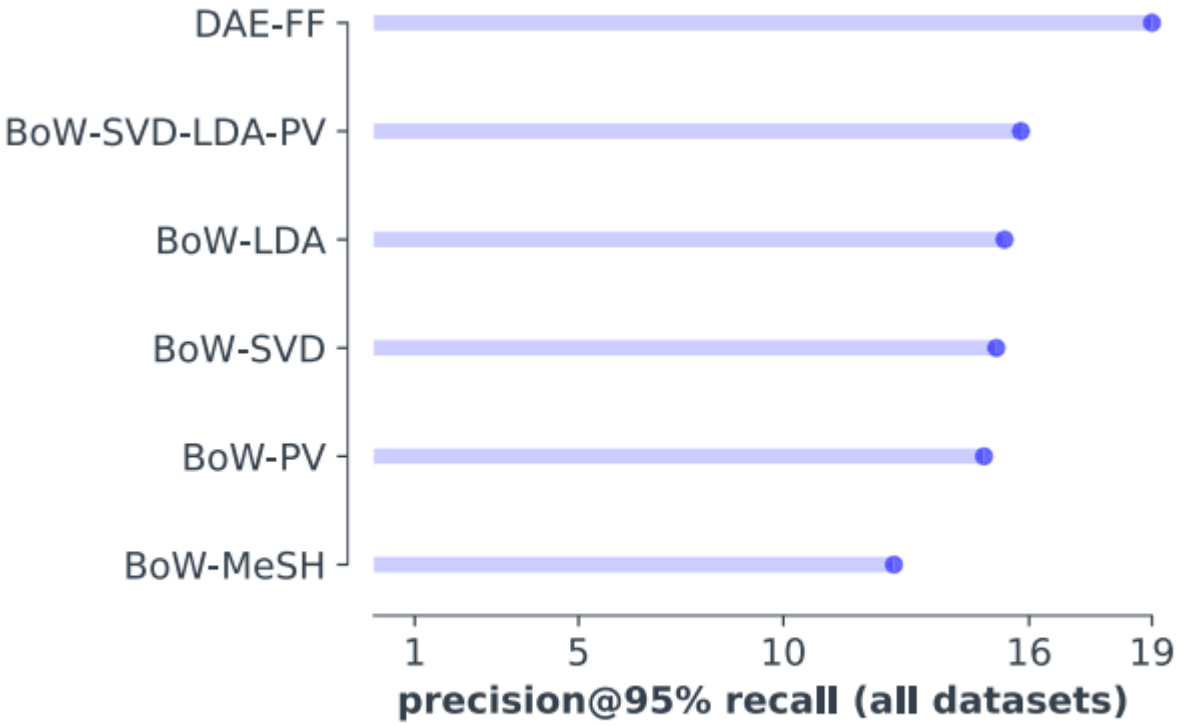



**Fig. 4.** Average (all datasets) precision@95% recall of our method against 5 composite feature extraction methods.

- Proposed method shows the best performance by outperforming the 5 composite feature extraction methods by 3.2% to 6.3%.
- The composite methods obtain approximately the same performance with the exception of the BoW-MeSH that shows a substantially lower average precision at recall level of 95% of ~ 13%.

# Experiments

## Discussion

1. The neural network-based feature extraction method substantially reduced the screening workload of 23 systematic reviews by approximately 56%.
2. The workload savings varied across the 23 reviews from a low WSS@95% score of ~9% on the Oral Hypoglycemics review to a higher WSS@95% score of ~84% on the PFOA/PFOS review.
3. A weak correlation ($R^2 = 0.279$) between the WSS@95% performance and the size of the corresponding review dataset which was statistically insignificant ($p = .197$).
    - This indicates that proposed method can obtain meaningful workload savings on both small and large review datasets.
4. The proposed feature extraction method yields significant workload savings of at least 10% in 22 out of 23 review datasets and thus it could be potentially used in practical application scenarios for accelerating the citation screening task of systematic reviews.

# Experiments

## Limitations

1. The minimum value of the ranked list that discriminates higher ranked eligible studies from lower ranked ineligible studies, is pre-defined and fixed at 95% recall.
    - In practical scenarios such a threshold value is difficult to define.
    - The optimal cut-off threshold varies greatly across different reviews.
2. The underlying neural network-based feature extraction method is trained independently for each systematic review dataset.
    - As an example, in our experiments we produced 23 neural network models corresponding to the 23 review datasets.

## Conclusion

1. Presented a text classification method to accelerate the citation screening process of systematic reviews.
    - The method aims to minimise the human workload involved in citation screening.
        - Human reviewers need to manually label only a subset of the citations.
        - The remaining unlabelled citations are automatically labelled by the text classification method.
2. Demonstrated by initialising the feed forward neural network using multiple denoising autoencoders of varying dimensionality we can improve upon the performance of our feature extraction method.

# Experiments

## Conclusion

3. Further performed a number of experiments to assess the performance of our method across 23 publicly available systematic review datasets.
    - It was shown that for 22 out of 23 review datasets the proposed method achieved significant workload savings on at least 10%.
    - In several cases our method yielded a statistically significantly better performance over 10 baseline feature extraction methods.

# Contributions

- New neural network-based feature extraction for text analysis.
- New model to discriminate between eligible and ineligible citations for screening process in systematic reviews.