

Introduction to the basics of AI

Session 6

Z. TAIA-ALAOUI

Outline

- Definitions
- SVM / MLP
- Shallow Neural Network
- Implementation

Statistical Tools - Dataset

- **Sample**

$$X_{k \in [1, N]} \in \mathbb{R}^p$$

- **Set of samples**

$$\mathbf{X} = \{X_k \in \mathbb{R}^p\}_{k \in [1, N]}$$

IRIS DATASET

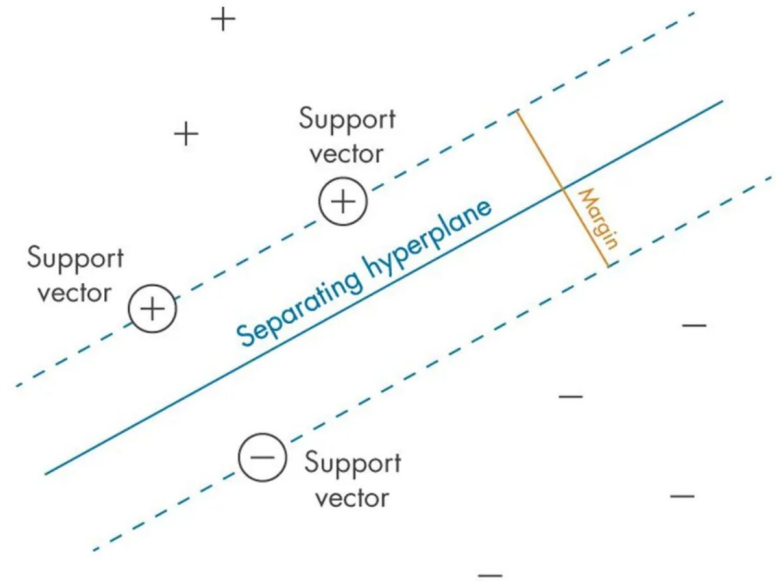
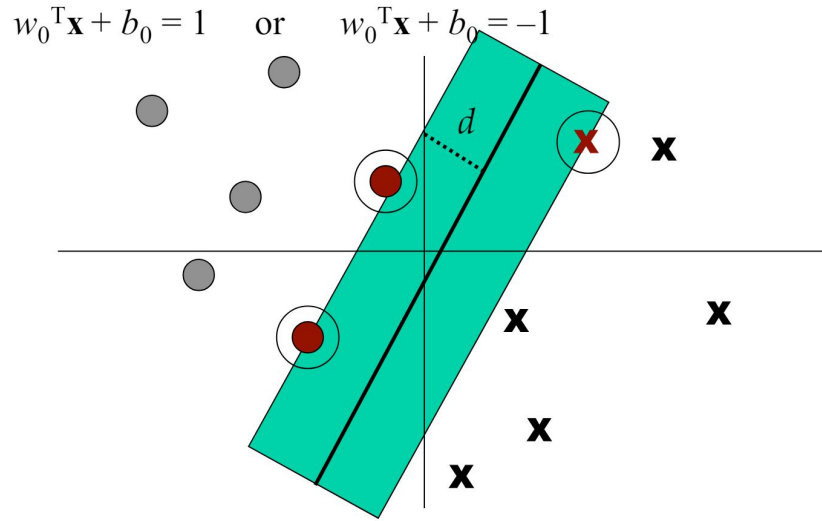
	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
1	5.1	3.5	1.4	0.2
2	4.9	3.0	1.4	0.2
3	4.7	3.2	1.3	0.2
4	4.6	3.1	1.5	0.2
5	5.0	3.6	1.4	0.2
6	5.4	3.9	1.7	0.4
7	4.6	3.4	1.4	0.3
8	5.0	3.4	1.5	0.2

Statistical Tools - Dataset

- **Set of N samples expressed in a space of p Variables**

$$\begin{aligned}
 \mathbf{X} &= \begin{pmatrix} x_{11} & x_{12} & x_{13} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \color{red}{X_i} & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ x_{N1} & \cdot & \cdot & \cdot & x_{Np} \end{pmatrix} = [V_1, V_2, \dots, V_p] = \begin{bmatrix} X_1^T \\ X_2^T \\ \cdot \\ \cdot \\ \cdot \\ X_N^T \end{bmatrix} \\
 & \quad \quad \quad X_i = \begin{bmatrix} x_{i1} \\ x_{i2} \\ \cdot \\ \cdot \\ \cdot \\ x_{ip} \end{bmatrix} \quad V_j = \begin{bmatrix} x_{1j} \\ x_{2j} \\ \cdot \\ \cdot \\ \cdot \\ x_{Nj} \end{bmatrix}
 \end{aligned}$$

SVM - Definitions



SVM - Definitions

- **Hyperplane:** Hyperplane is the decision boundary that is used to separate the data points of different classes in a feature space.
- **Support Vectors:** Support vectors are the closest data points to the hyperplane.
- **Margin:** Margin is the distance between the support vector and hyperplane.
- **Kernel:** Kernel is the mathematical function, which is used in SVM to map the original input data points into high-dimensional feature spaces.
- **C:** Margin maximisation and misclassification fines are balanced by the regularisation parameter C in SVM. A stricter penalty is imposed with a greater value of C , which results in a smaller margin and perhaps fewer misclassifications

SVM - Definitions

Hard Margin

$$\hat{y} = \begin{cases} 1 & : w^T x + b \geq 0 \\ 0 & : w^T x + b < 0 \end{cases}$$

$$\begin{aligned} & \underset{w,b}{\text{minimize}} \frac{1}{2} w^T w = \underset{W,b}{\text{minimize}} \frac{1}{2} \|w\|^2 \\ & \text{subject to } y_i(w^T x_i + b) \geq 1 \text{ for } i = 1, 2, 3, \dots, m \end{aligned}$$

Soft Margin

$$\begin{aligned} & \underset{w,b}{\text{minimize}} \frac{1}{2} w^T w + C \sum_{i=1}^m \zeta_i \\ & \text{subject to } y_i(w^T x_i + b) \geq 1 - \zeta_i \text{ and } \zeta_i \geq 0 \text{ for } i = 1, 2, 3, \dots, m \end{aligned}$$

SVM - Definitions

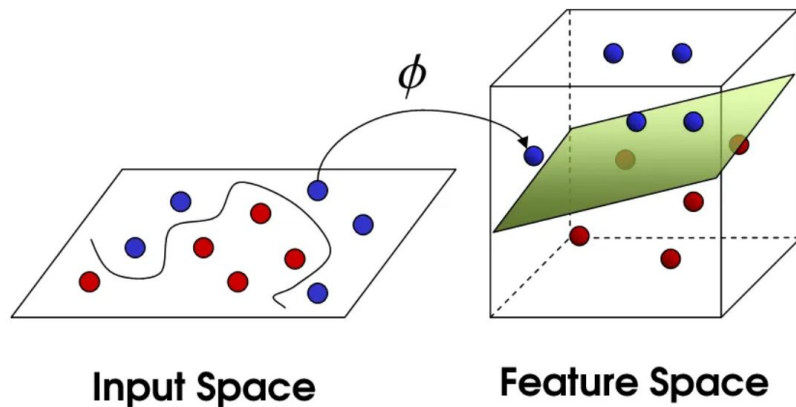
- Non-linear case

Linear : $K(w, b) = w^T x + b$

Polynomial : $K(w, x) = (\gamma w^T x + b)^N$

Gaussian RBF: $K(w, x) = \exp(-\gamma \|x_i - x_j\|^n)$

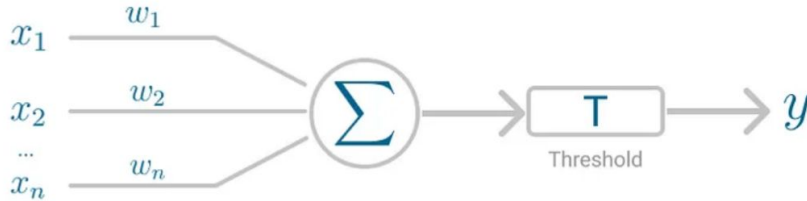
Sigmoid : $K(x_i, x_j) = \tanh(\alpha x_i^T x_j + b)$



Multi-Layer Perceptron - Definitions

- The perceptron imitates human perception

$$\underbrace{f(x, w)}_{\text{output}} = \underbrace{x_1 w_1 + \cdots + x_n w_n}_{\substack{\text{inputs} \\ \text{weights}}}$$



$$\hat{y} = \text{sign}(w^T x + b)$$

- T is called activation function

Multi-Layer Perceptron - Definitions

- Classification with MLP

$$L_{\text{Perceptron}}(w, b) = \sum_{i=1}^M \max(0, -y^{(i)}(w^T x^{(i)} + b))$$

$$w^T x + b < 0$$



Mis-classified sample

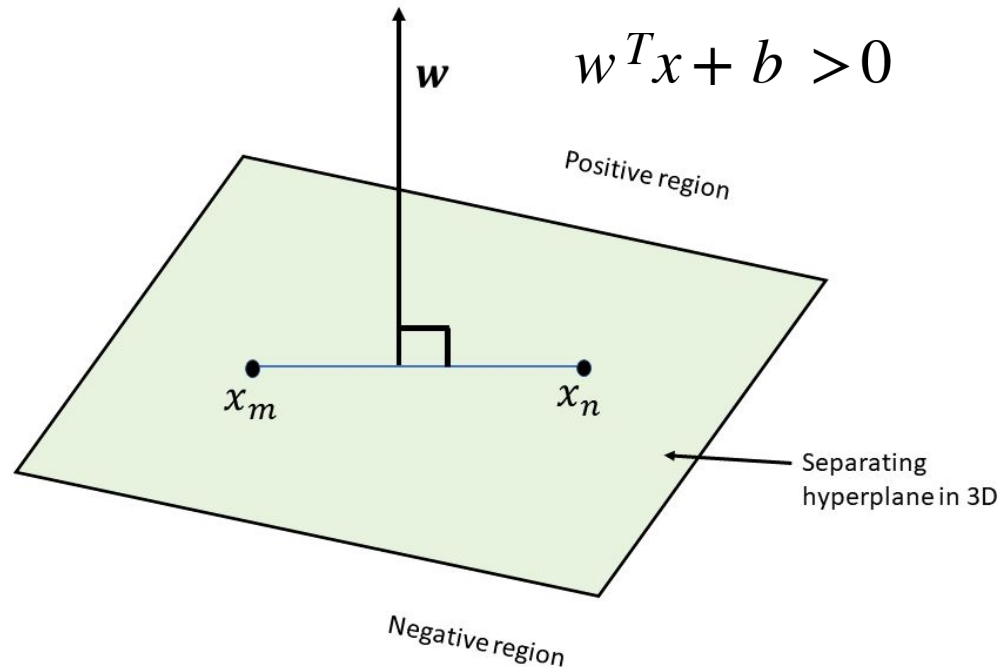
$$w^T x + b > 0$$



Well classified sample

Multi-Layer Perceptron - Definitions

- Classification with MLP: Geometrical Interpretation



Multi-Layer Perceptron - Definitions

- Classification with MLP: Learning

Data: Training Data: $(x_i, y_i); \forall i \in \{1, 2, \dots, N\}$, Learning Rate: η

Result: Separating Hyper-plane coefficients : \mathbf{w}^*

Initialize $\mathbf{w} \leftarrow \mathbf{0}$;

repeat

 get example (x_i, y_i) ;

$\hat{y}_i \leftarrow w^T x_i$;

if $\hat{y}_i y_i \leq 0$ **then**

$w \leftarrow w + \eta y_i x_i$

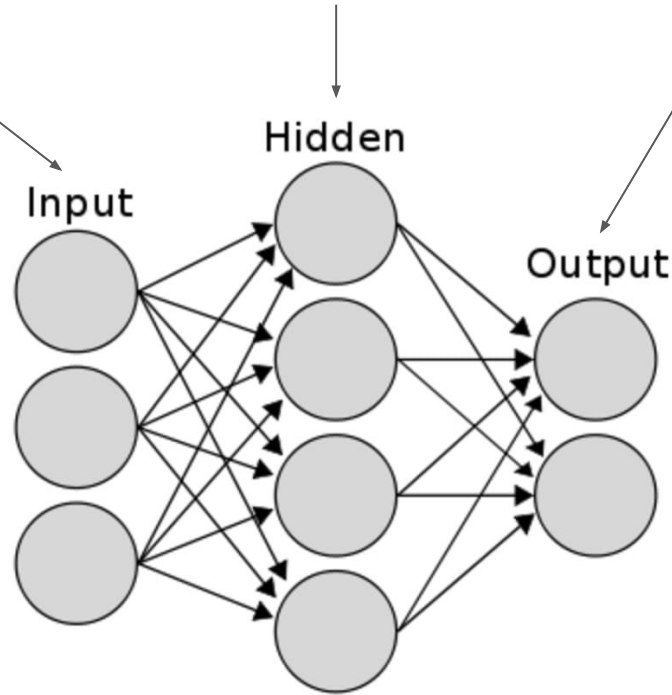
until *convergence*;

Vanilla (shallow) NN

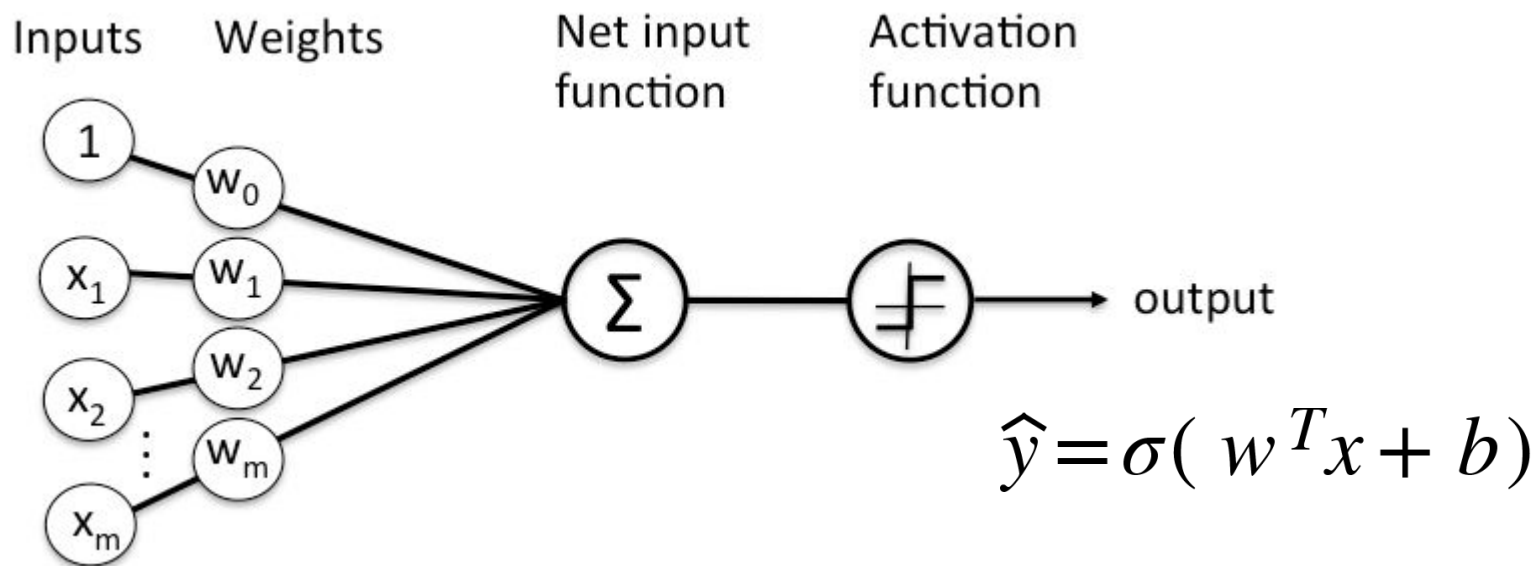
Hidden Feature Space (Latent space)

Your measurements

Your desired outputs



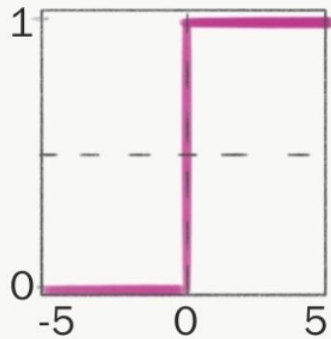
ANN



Activation Functions

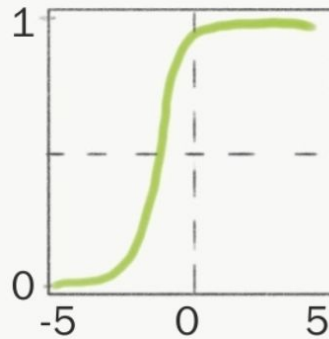
Activation Functions

Threshold

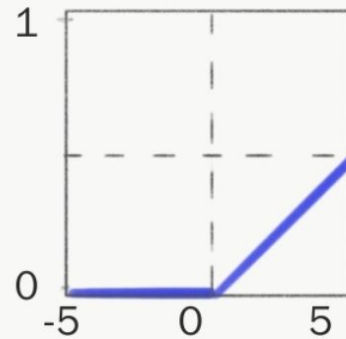


No defined
gradient

Sigmoid

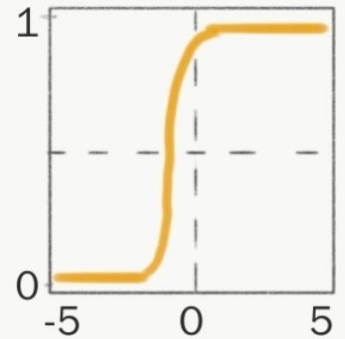


Relu*



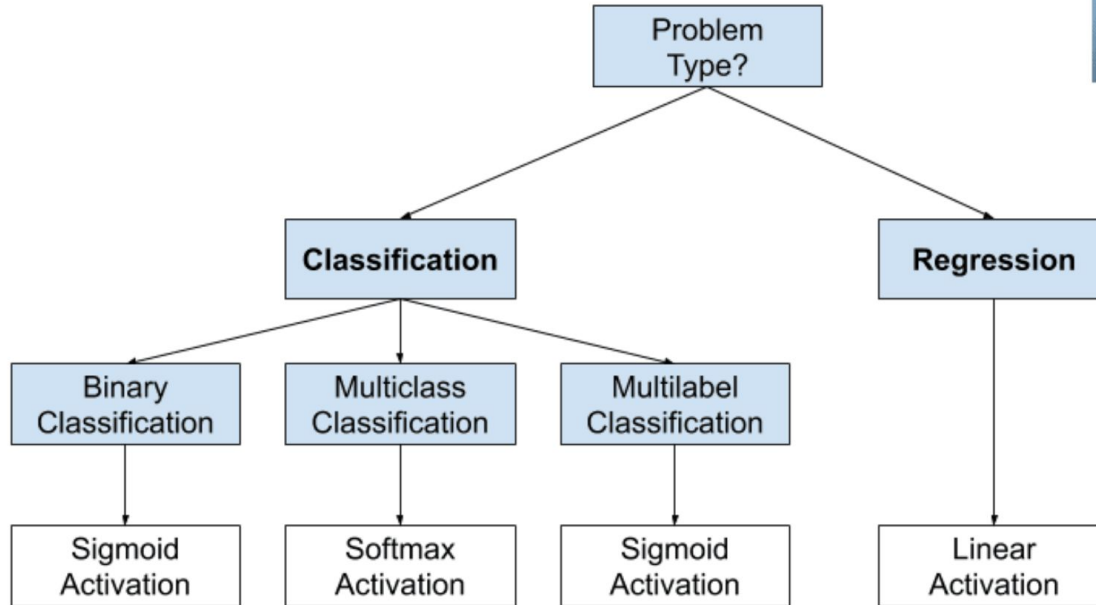
Default

Tan H



Activation Functions

How to Choose an Output Layer Activation Function



MachineLearningMastery.com

How to Choose an Output Layer Activation Function

Activation Functions

Sigmoid

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Softmax

$$s(x_i) = \frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}}$$

Learning

- Forward Propagation

$$\mathbf{z} = \mathbf{W}^{(1)}\mathbf{x}$$

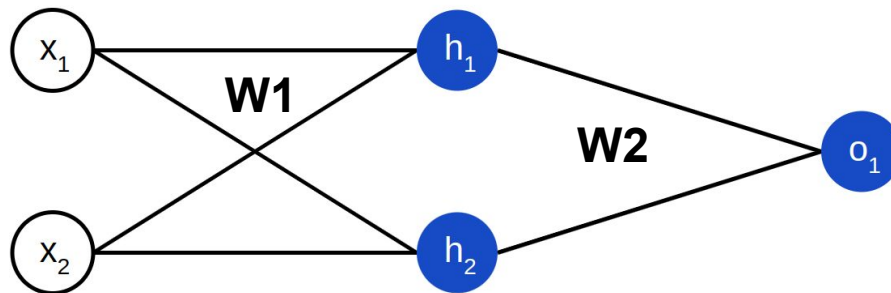
$$\mathbf{h} = \phi(\mathbf{z})$$

$$\mathbf{o} = \mathbf{W}^{(2)}\mathbf{h}$$

Input Layer

Hidden Layer

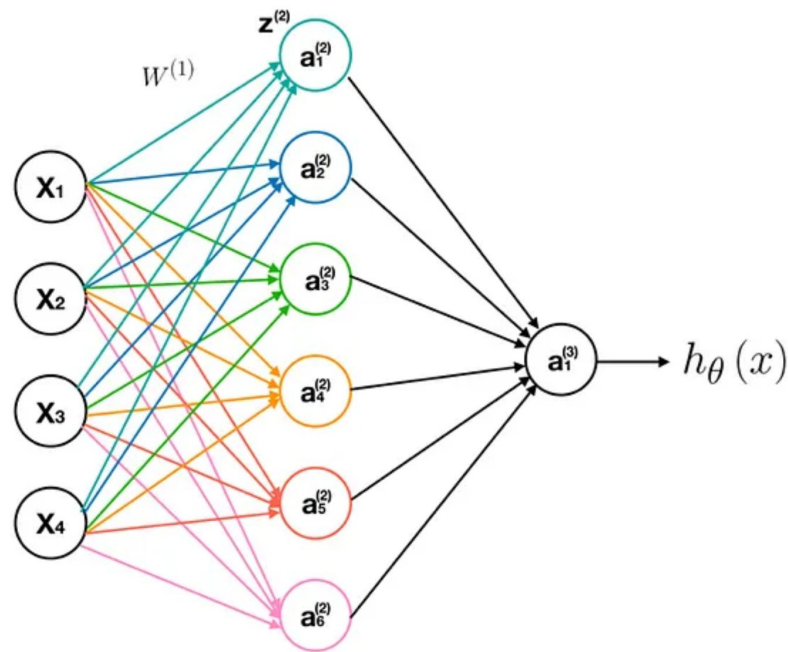
Output Layer



Learning

- Forward Propagation

$$W^T X = \begin{bmatrix} \theta_{11}^{(1)} & \theta_{12}^{(1)} & \theta_{13}^{(1)} & \theta_{14}^{(1)} \\ \theta_{21}^{(1)} & \theta_{22}^{(1)} & \theta_{23}^{(1)} & \theta_{24}^{(1)} \\ \theta_{31}^{(1)} & \theta_{32}^{(1)} & \theta_{33}^{(1)} & \theta_{34}^{(1)} \\ \theta_{41}^{(1)} & \theta_{42}^{(1)} & \theta_{43}^{(1)} & \theta_{44}^{(1)} \\ \theta_{51}^{(1)} & \theta_{52}^{(1)} & \theta_{53}^{(1)} & \theta_{54}^{(1)} \\ \theta_{61}^{(1)} & \theta_{62}^{(1)} & \theta_{63}^{(1)} & \theta_{64}^{(1)} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} = \begin{bmatrix} z_1^{(1)} \\ z_2^{(1)} \\ z_3^{(1)} \\ z_4^{(1)} \\ z_5^{(1)} \\ z_6^{(1)} \end{bmatrix} = Z^{(2)}$$



Learning

- Backward Propagation

$$\mathbf{z} = \mathbf{W}^{(1)}\mathbf{x}$$

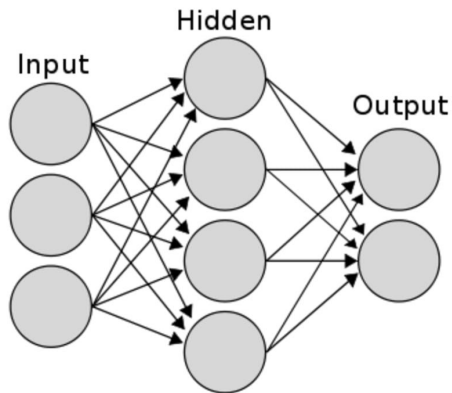
$$\mathbf{h} = \phi(\mathbf{z})$$

$$\mathbf{o} = \mathbf{W}^{(2)}\mathbf{h}$$

Loss

$$L = l(\mathbf{o}, y)$$

$$\hat{o} = \sigma(w^T x + b)$$



$$w_{ij}^{(k)} = w_{ij}^{(k)} - \eta \frac{\partial L}{\partial w_{ij}^{(k)}}$$

$$b_i^{(k)} = b_i^{(k)} - \eta \frac{\partial L}{\partial b_i^{(k)}}$$

Vocabulary

- Batch Learning: for each epoch, a batch of samples is select to train the model
- Epoch: complete sequence of iterations in a batch
- Iteration: one backward pass
- Learning rate: speed at which weights are updated

Sources

<https://towardsdatascience.com/multilayer-perceptron-explained-with-a-real-life-example-and-python-code-sentiment-analysis-cb408ee93141>

<https://web.mit.edu/6.034/wwwbob/svm-notes-long-08.pdf>

https://course.ccs.neu.edu/cs6140sp15/2_GD_REG_pton_NN/lecture_notes/lectureNotes_Perceptron.pdf

<https://web.mit.edu/6.034/wwwbob/svm-notes-long-08.pdf>

<https://towardsdatascience.com/the-kernel-trick-c98cdbcaeb3f>