



Music Data Mining

Introduction to Deep Learning

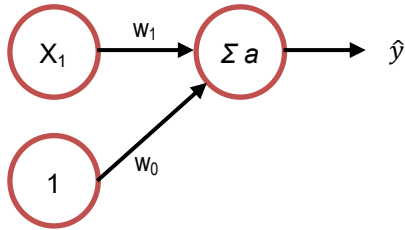
Topics

- Regression models in graphs
- Perceptron
- Shallow Neural Networks (SNN)
- Deep Neural Networks (DNN)



Regression Models in Graphs

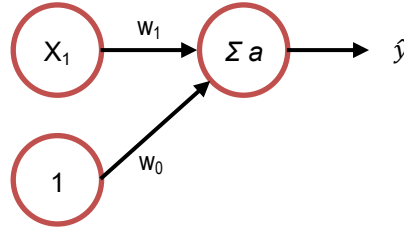
Linear Regression



$$a(w_1x_1 + w_0) = \hat{y}$$

a is the identity function

Logistic Regression



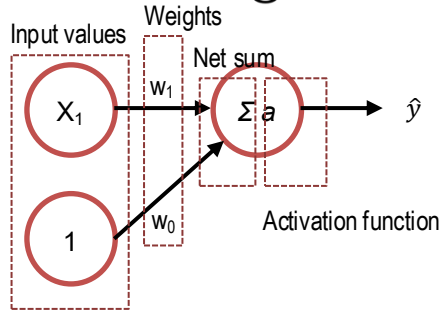
$$a(w_1x_1 + w_0) = \hat{y}$$

a is the sigmoid function



Regression Models in Graphs

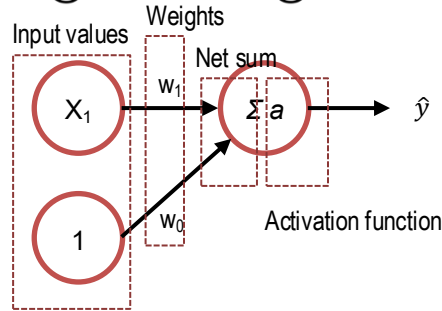
Linear Regression



$$a(w_1x_1 + w_0) = \hat{y}$$

a is the identity function

Logistic Regression

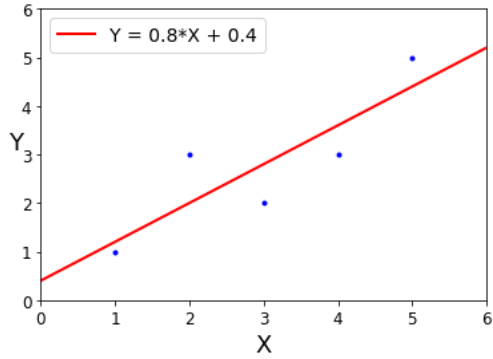


$$a(w_1x_1 + w_0) = \hat{y}$$

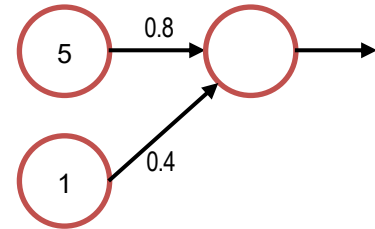
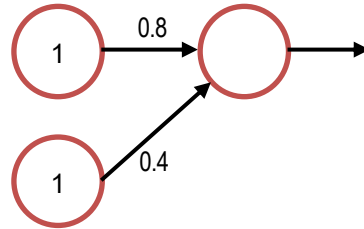
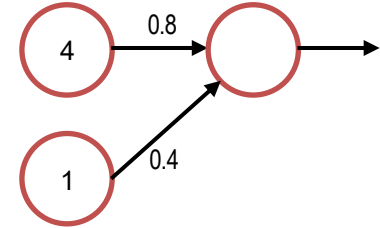
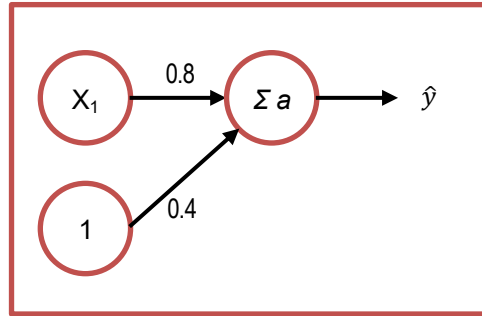
a is the sigmoid function



What are the Outputs?

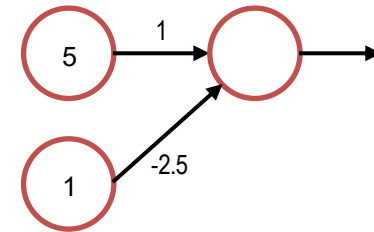
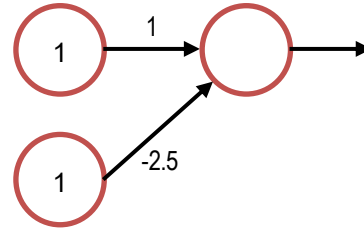
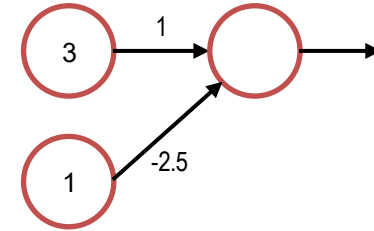
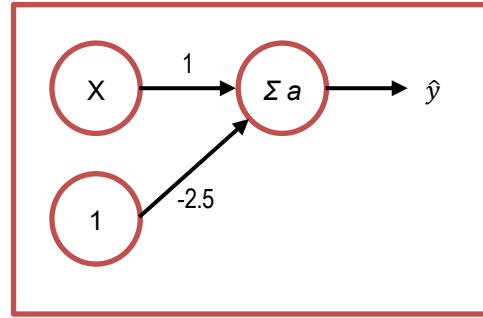
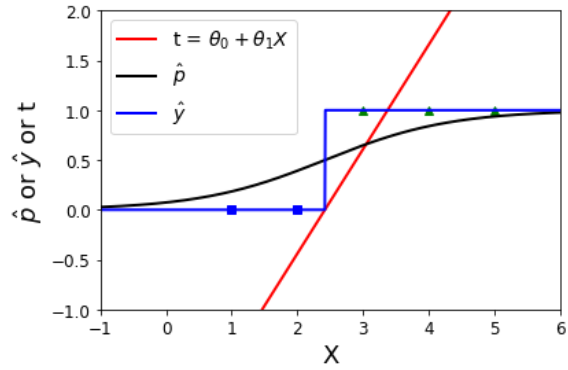


$a(0.8x_1 + 0.4) = \hat{y}$
a is the identity function



What are the Outputs?

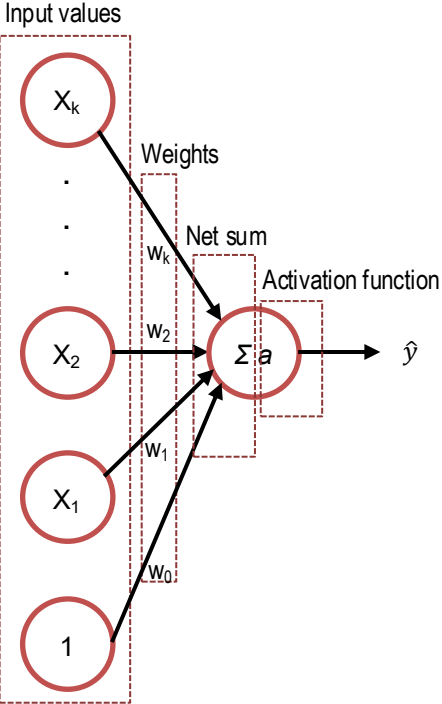
$$\begin{aligned}\sigma(-2.5) &= 0.08 \\ \sigma(-1.5) &= 0.18 \\ \sigma(0.5) &= 0.62 \\ \sigma(2.5) &= 0.92\end{aligned}$$



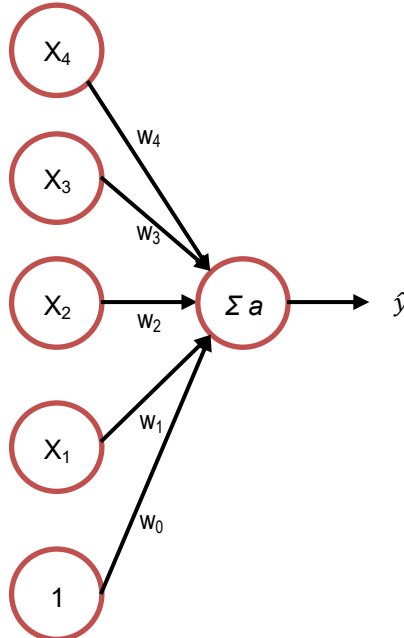
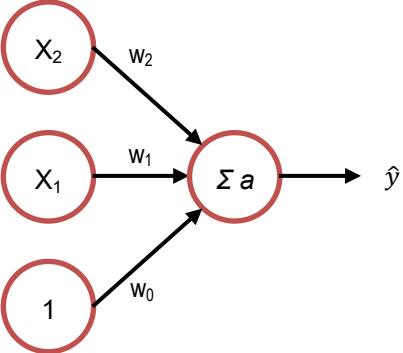
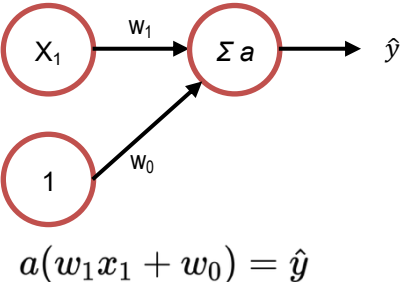
$a(x_1 - 2.5) = \hat{y}$
 a is the sigmoid function
and the prediction function



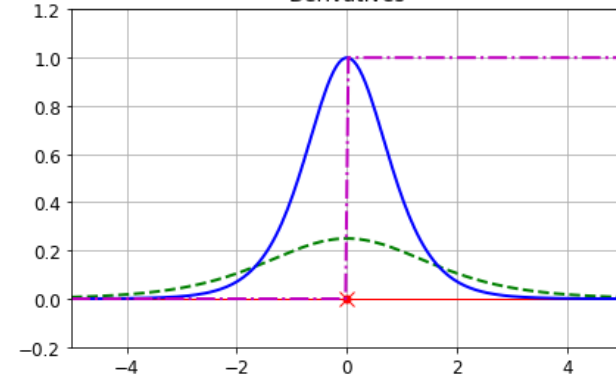
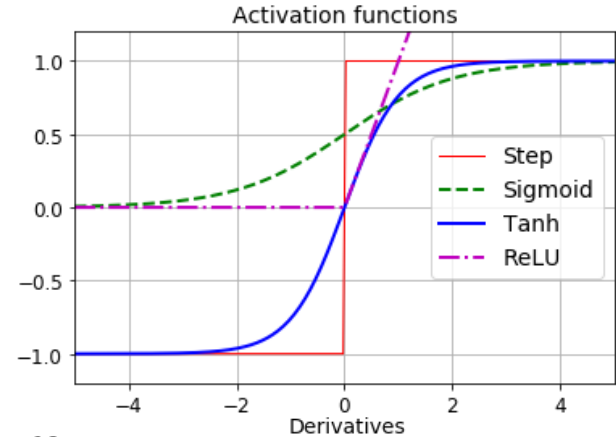
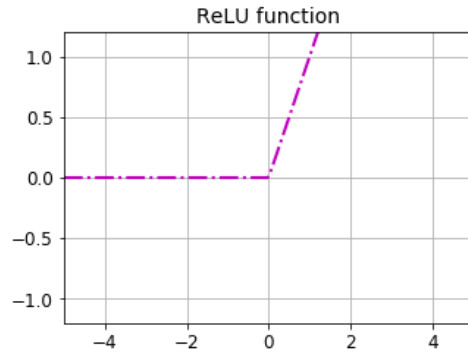
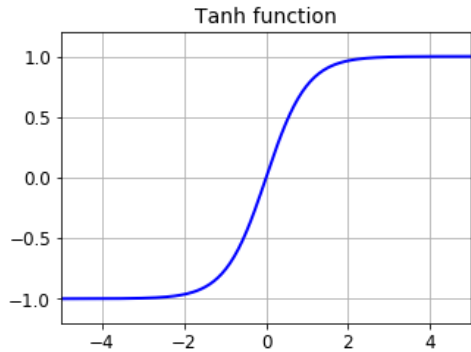
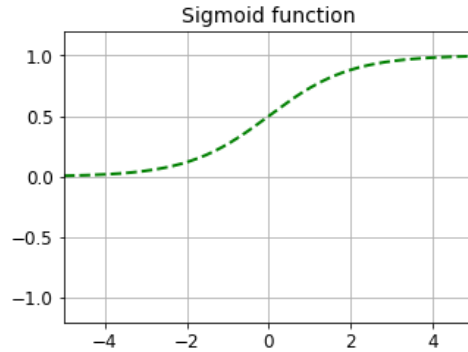
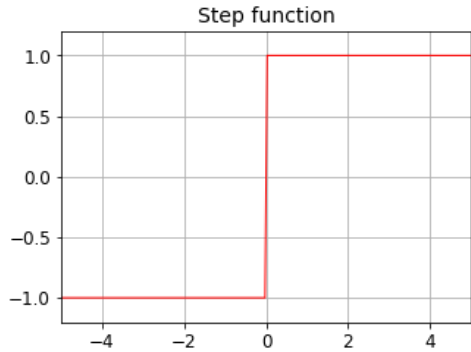
Perceptron



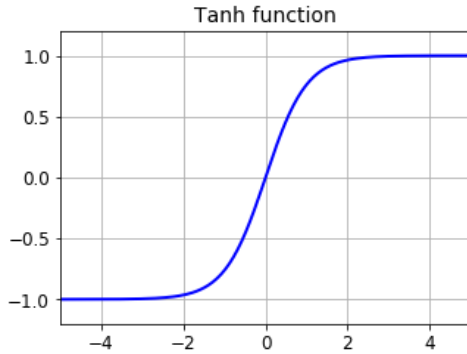
$$a(W^T X) = \hat{y}$$



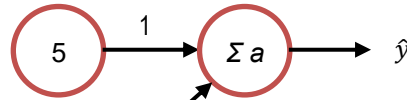
Activation Functions



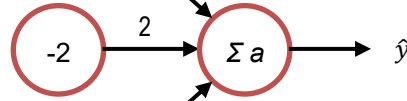
What are the Outputs?



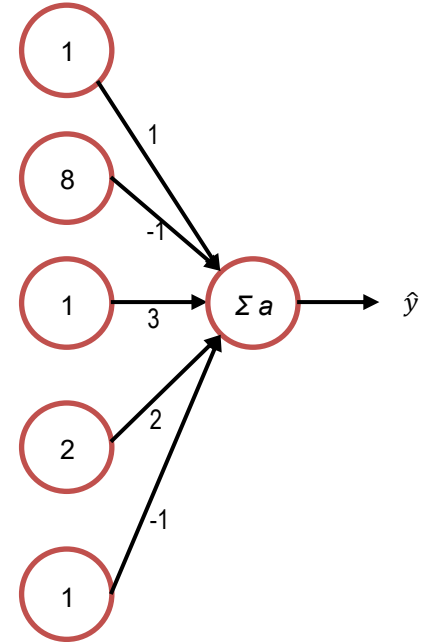
The activation function is the tanh function



$$a(x_1 - 5) = \hat{y}$$



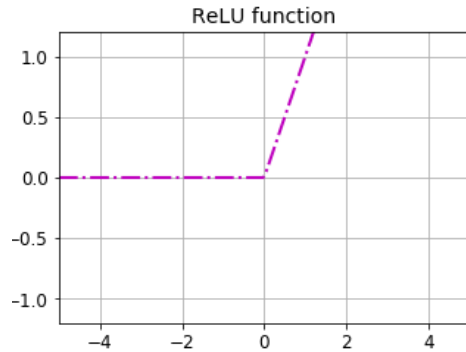
$$a(x_2 + 2x_0 + 3) = \hat{y}$$



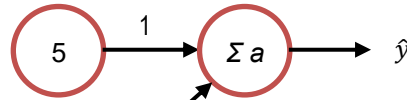
$$a(x_4 - x_3 + 3x_2 + 2x_0 - 1) = \hat{y}$$



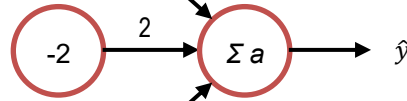
What are the Outputs?



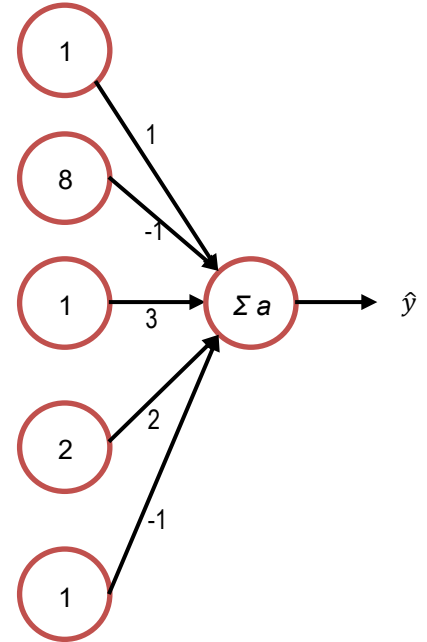
The activation function is the ReLU function



$$a(x_1 - 5) = \hat{y}$$



$$a(x_2 + 2x_0 + 3) = \hat{y}$$

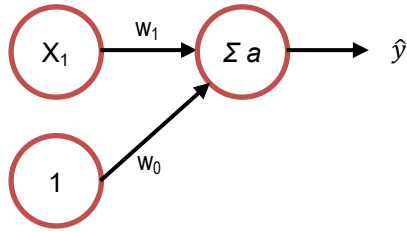


$$a(x_4 - x_3 + 3x_2 + 2x_0 - 1) = \hat{y}$$

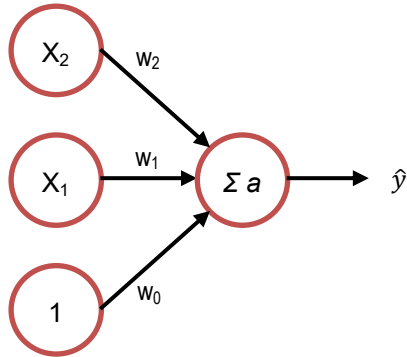


Deep Neural Network (DNN)

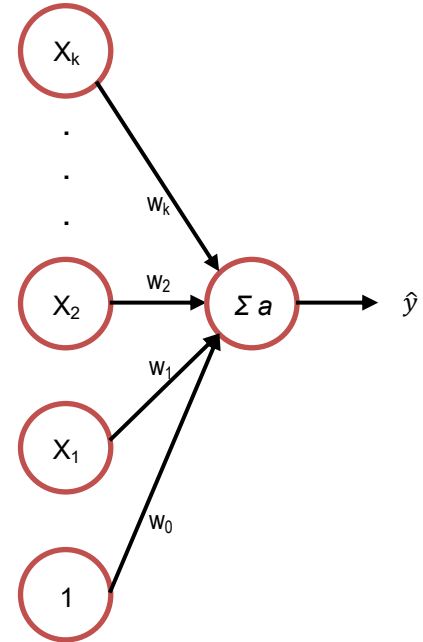
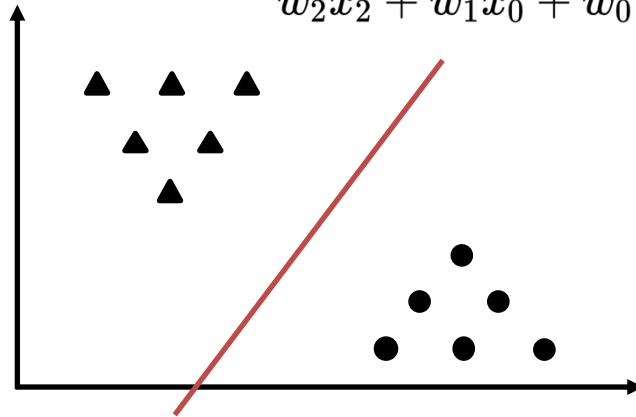
Perceptron for Linearly Separable Cases



$$w_1 x_0 + w_0 = 0$$

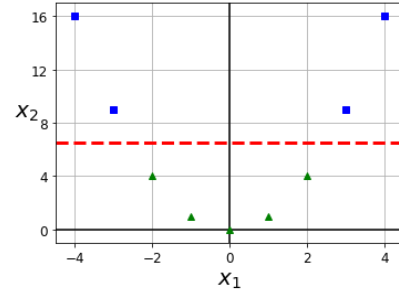
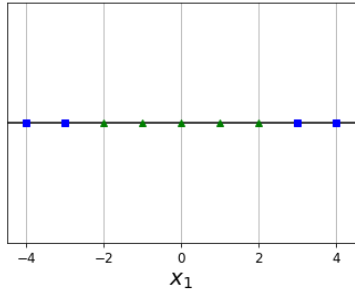


$$w_2 x_2 + w_1 x_0 + w_0 = 0$$

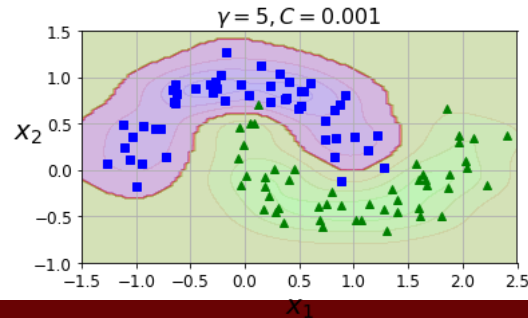
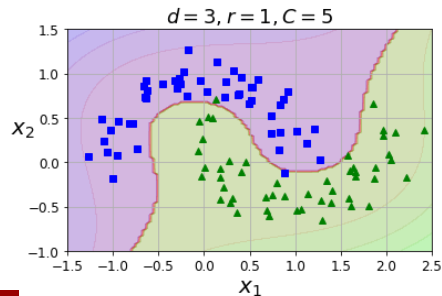


Not Linearly Separable Classification Problem

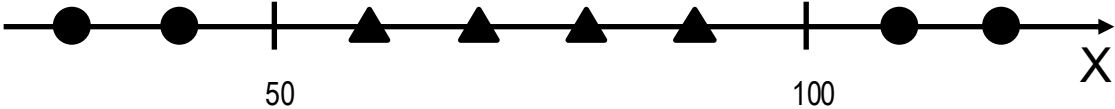
- Adding feature $x_2 = (x_1)^2$, to make a dataset linearly separable



- SVM Kernels: Polynomial kernel, RBF kernel



Not Linearly Separable Classification Problem

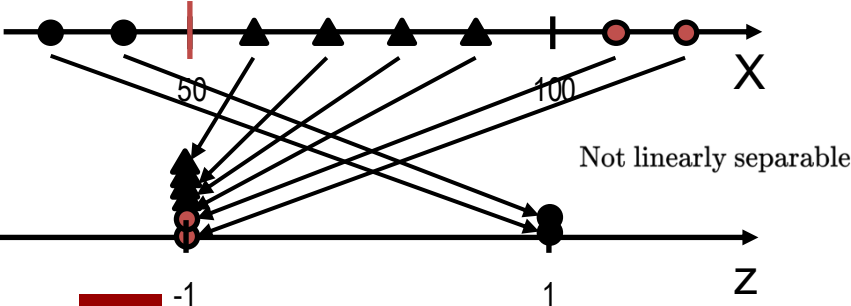


Feature transformation

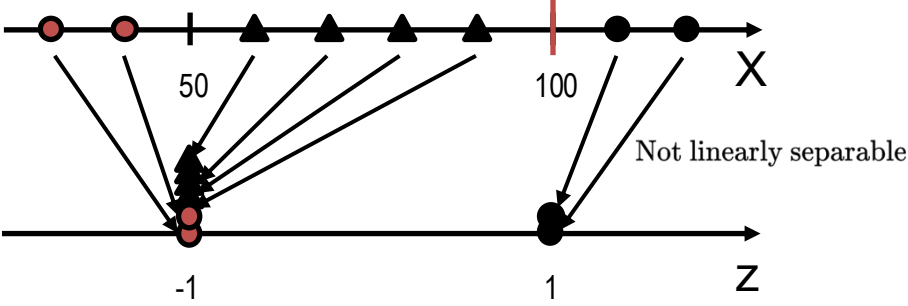
$$z = a(w_1x + w_0)$$

↑
Nonlinear functions (sigmoid, tanh, ReLU, etc.)

$$z = \text{sign}(w_1x + w_0) = \text{sign}(-x + 50)$$

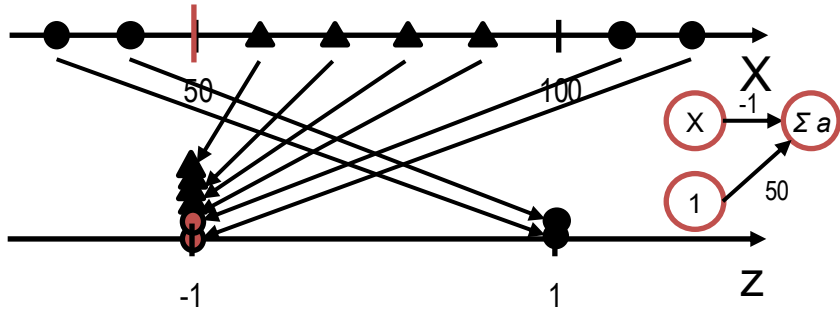


$$z = \text{sign}(w_1x + w_0) = \text{sign}(x - 100)$$

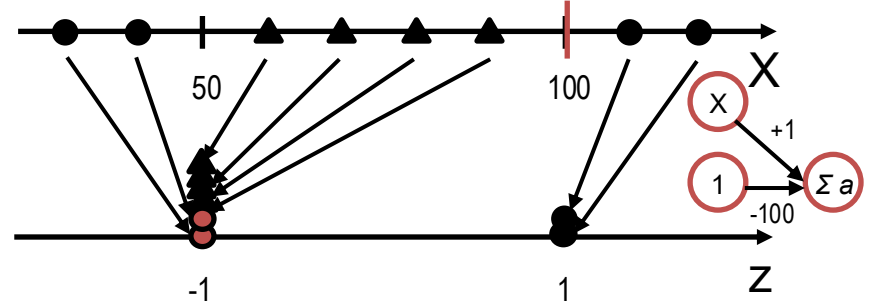


Not Linearly Separable Classification Problem

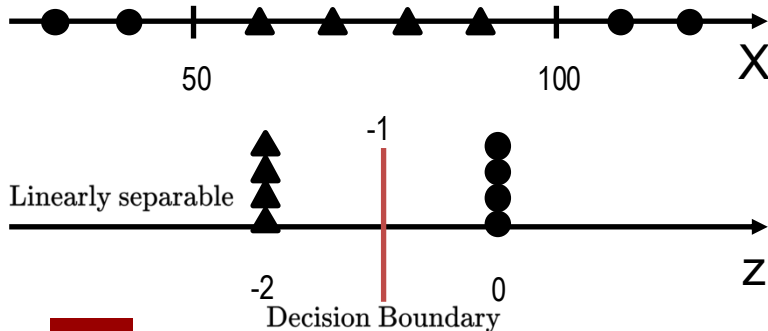
$$z = \text{sign}(w_1 x + w_0) = \text{sign}(-x + 50)$$



$$z = \text{sign}(w_1 x + w_0) = \text{sign}(x - 100)$$



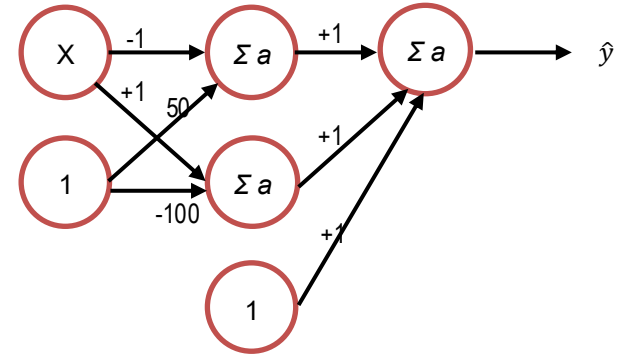
$$z = \text{sign}(-x + 50) + \text{sign}(x - 100)$$



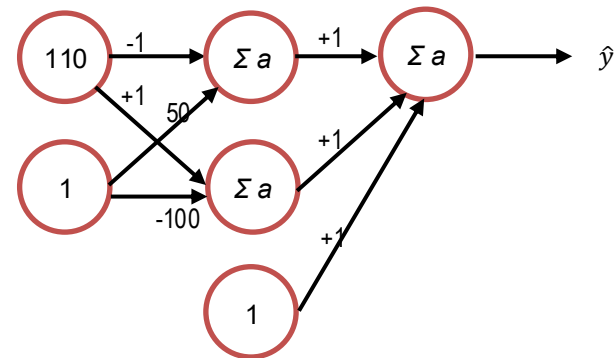
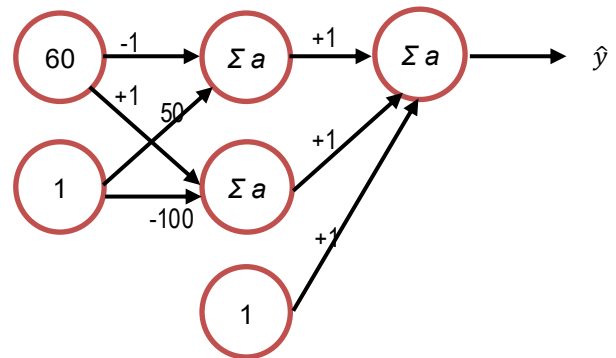
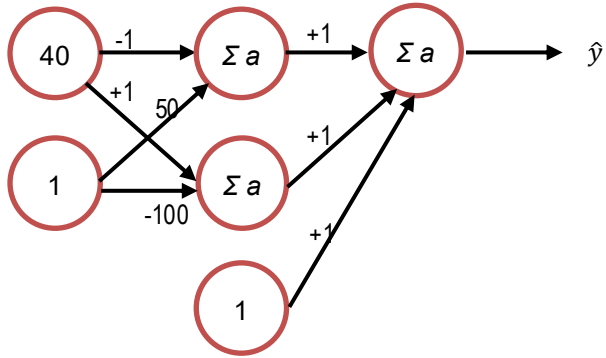
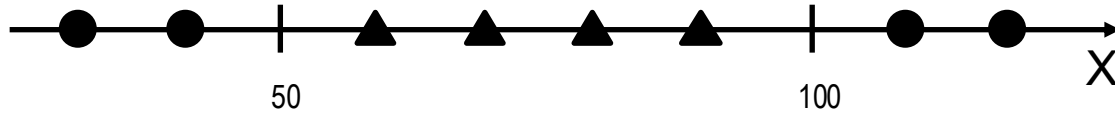
Hyperplane

$$w_1^{(2)} z + w_0^{(2)} = 0$$

$$w_1^{(2)} = 1, w_0^{(2)} = -1$$



Not Linearly Separable Classification Problem

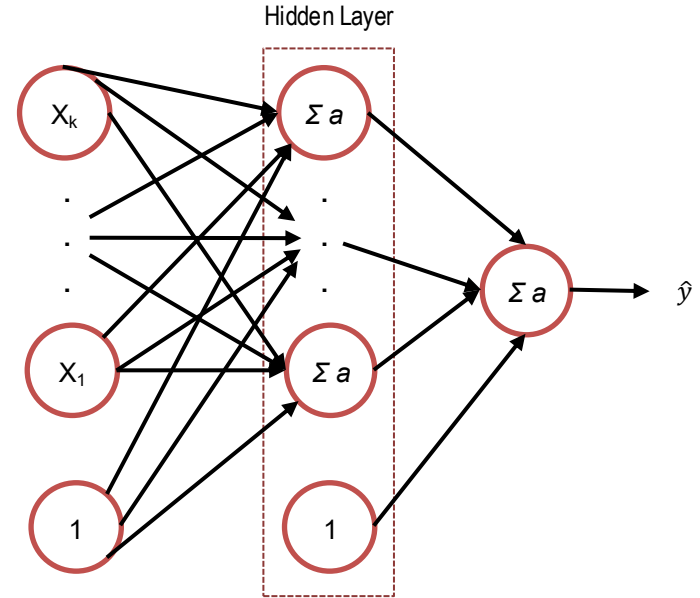


The activation function is the sign function

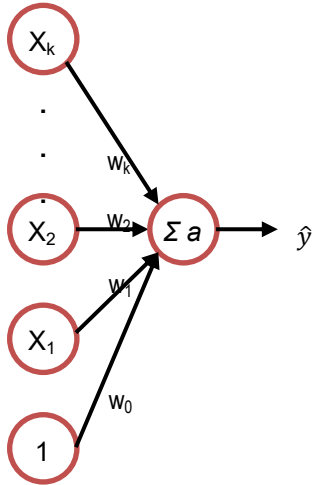


Shallow Neural Network (SNN)

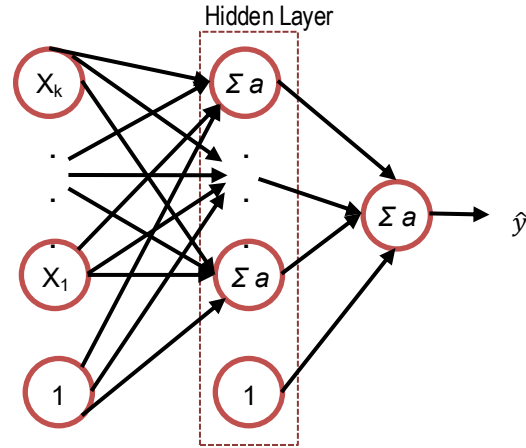
- SNN is a perceptron that takes perceptron outputs as inputs
 - SNN has one hidden layer
- Universal approximation theorem
 - SNN can approximate any continuous function



Perceptron vs. Shallow Neural Networks



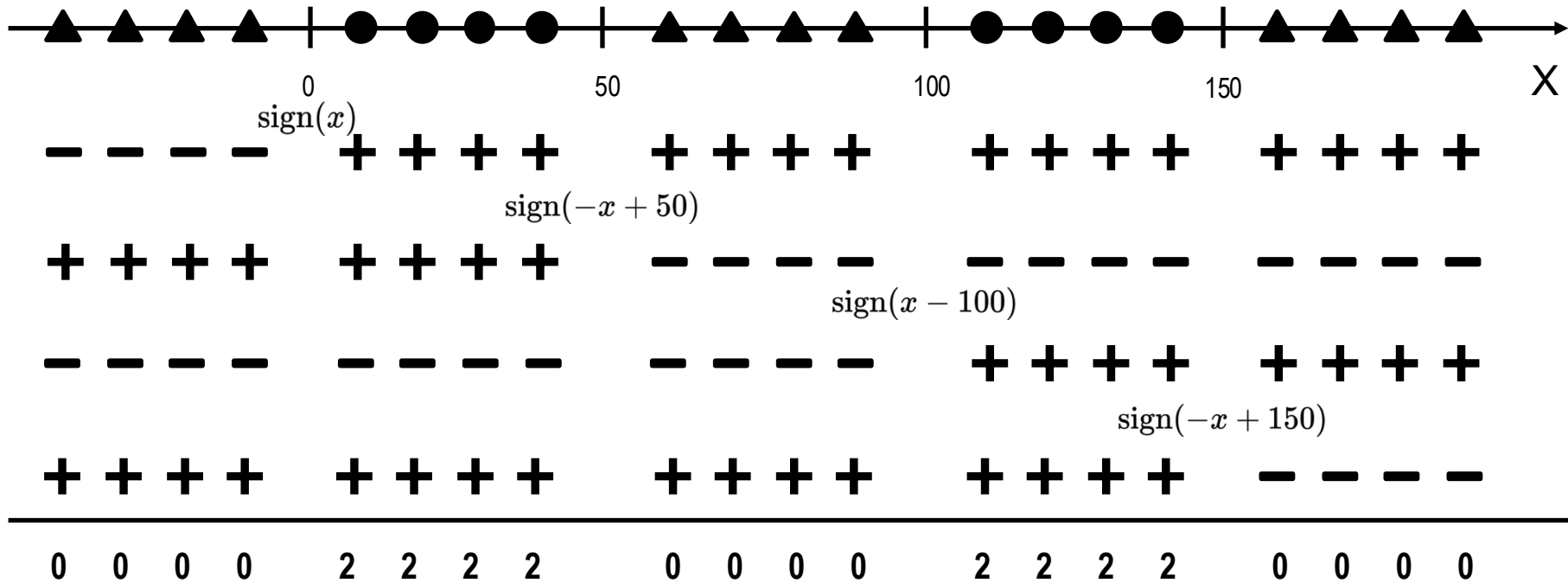
Only capable of learning linear problems



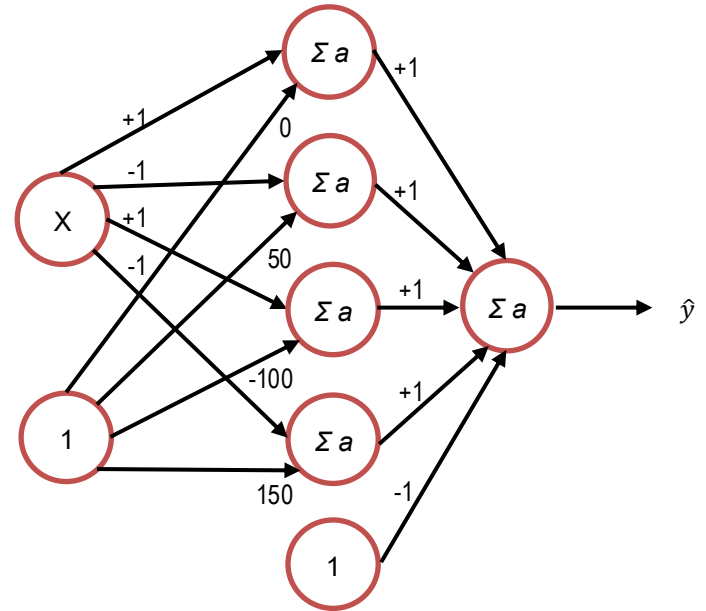
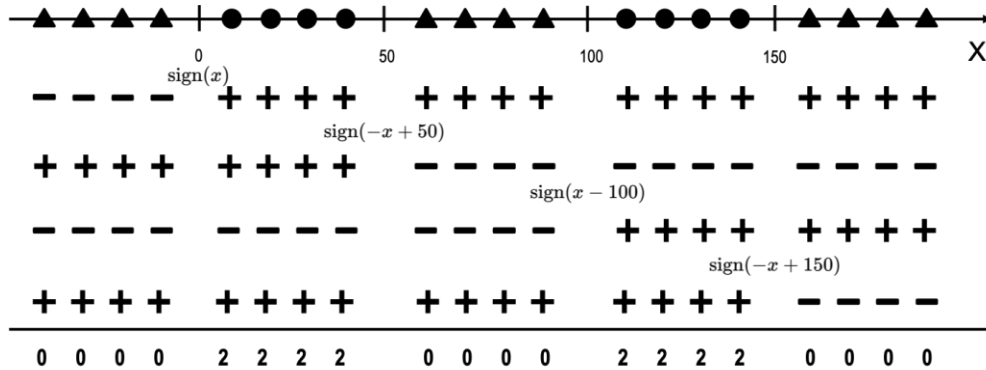
Capable of learning non-linear problems



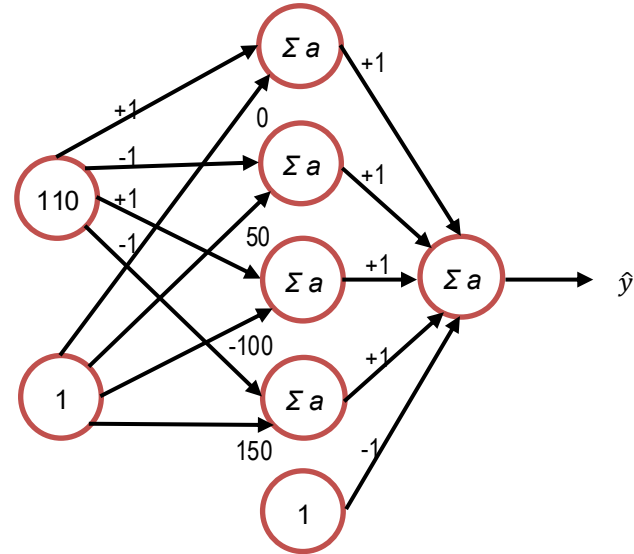
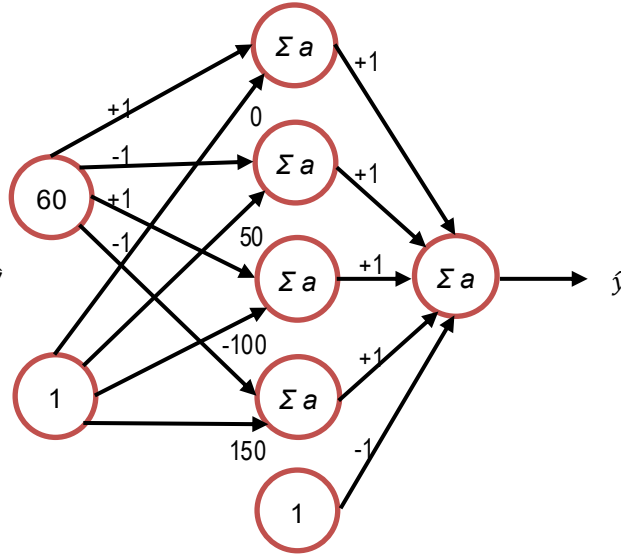
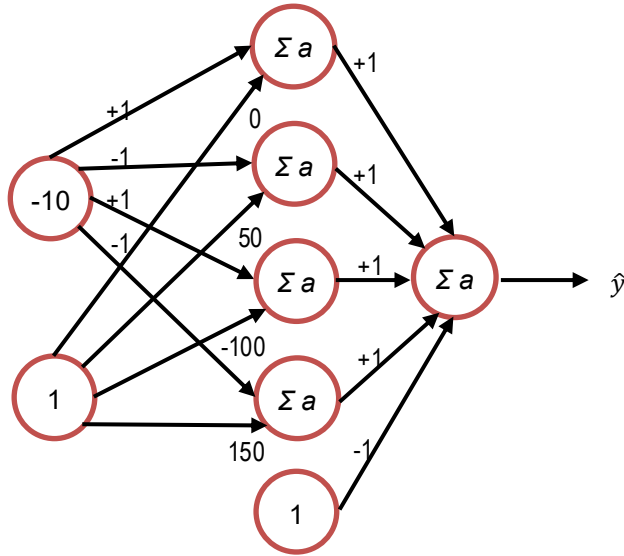
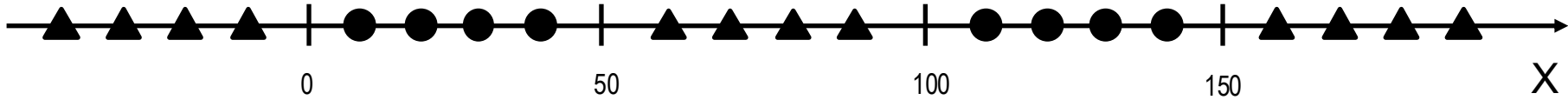
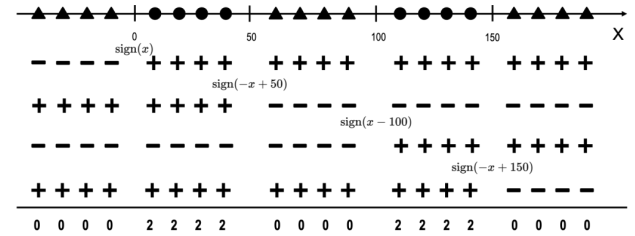
More Complex Example



More Complex Example

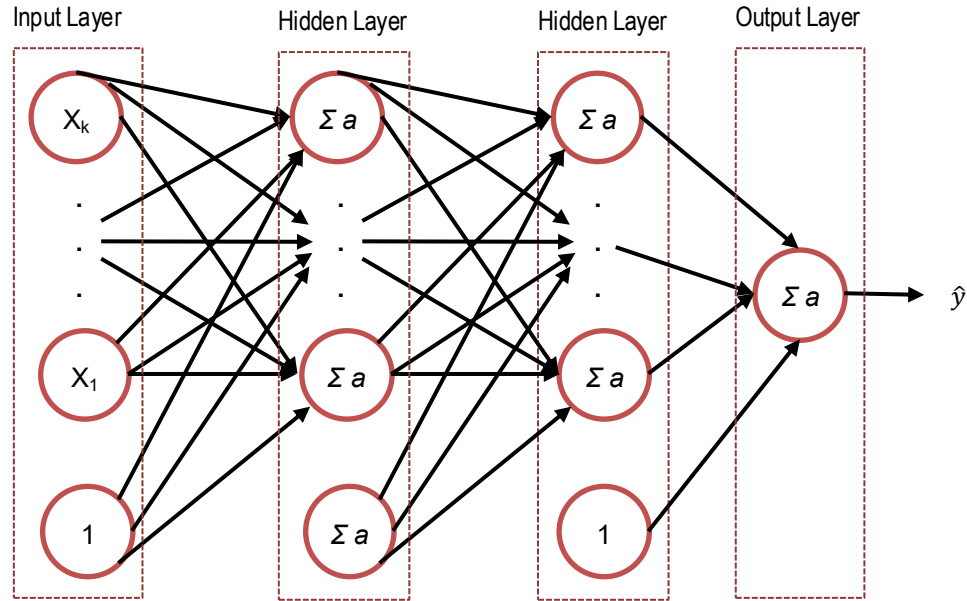


More Complex Example

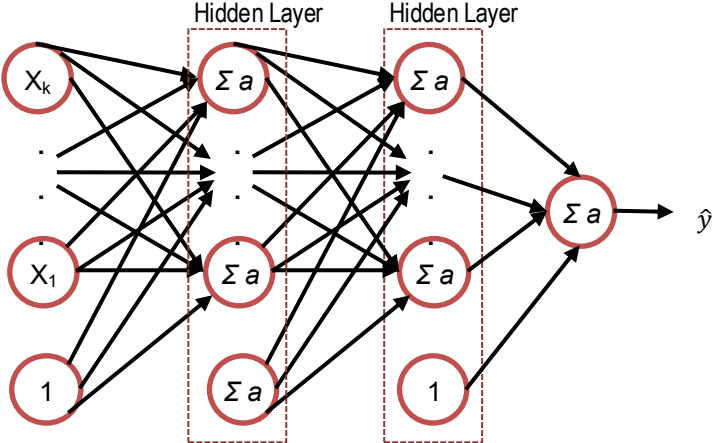
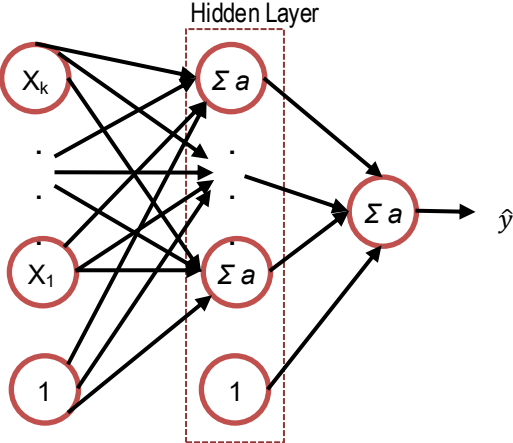
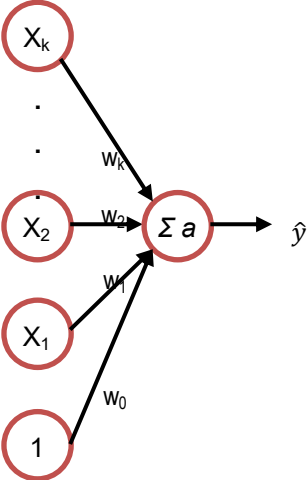


Deep Neural Networks (DNN)

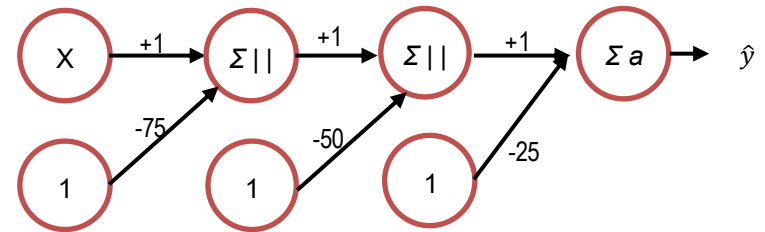
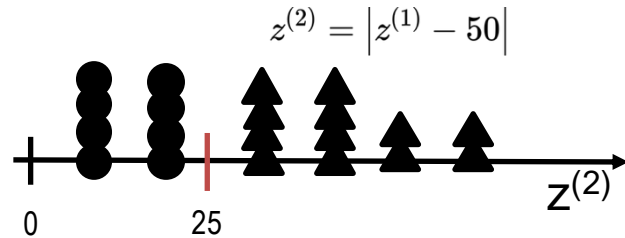
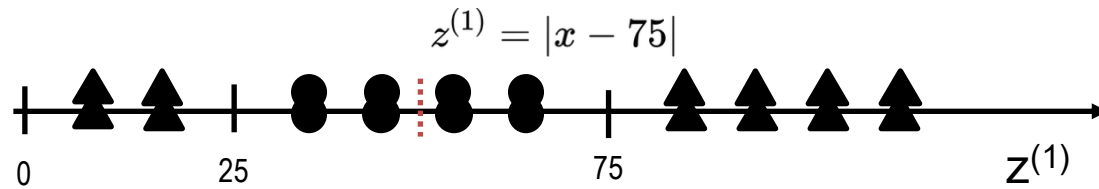
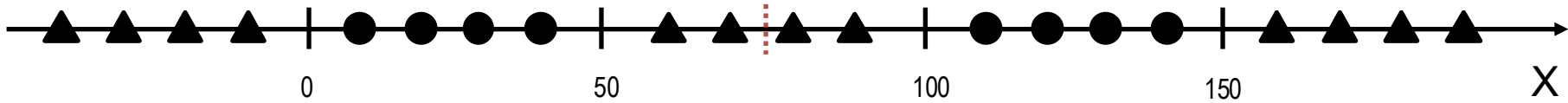
- DNN is a NN with many hidden layers
- DNN often outperforms SNN



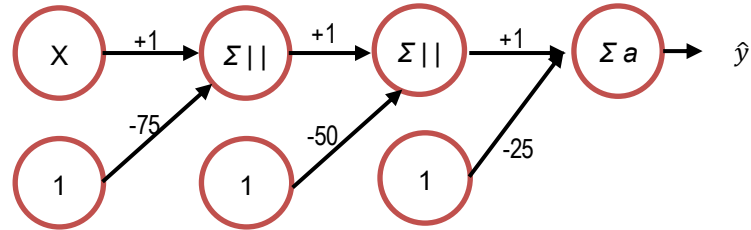
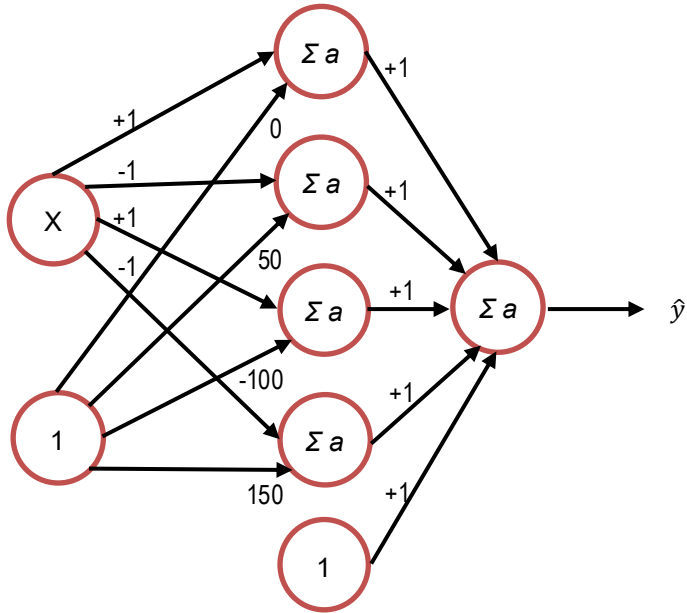
Perceptron vs. SNN vs. DNN



More Complex Example with DNN



SSN vs. DNN



- Deep networks internally build representations of patterns in the data
- Partially replace the need for feature engineering



DNN Learns Hierarchical Features

