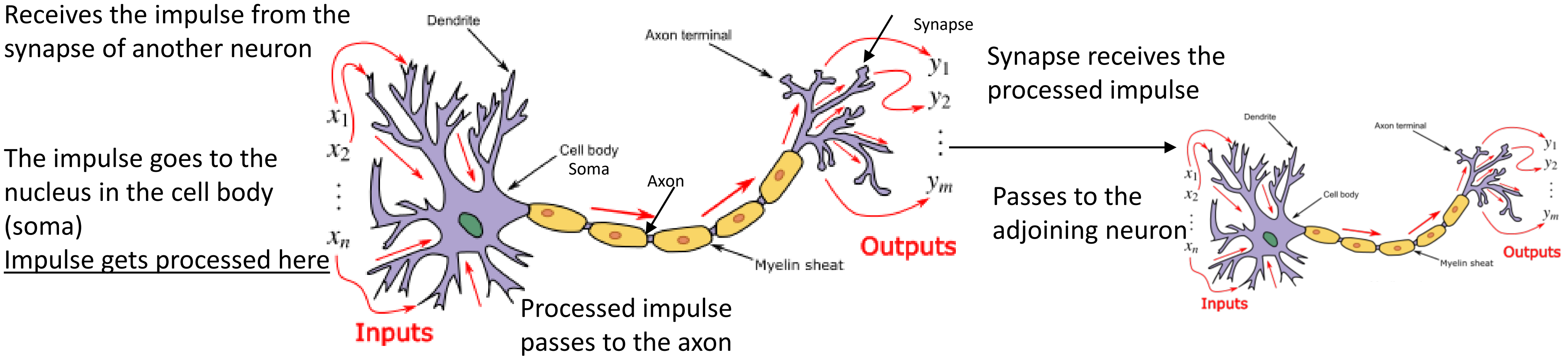
Nonlinear Models (Neural Networks)

Spring 2020

Instructor: Ankit Shah, Ph.D.

Biological Neural Networks



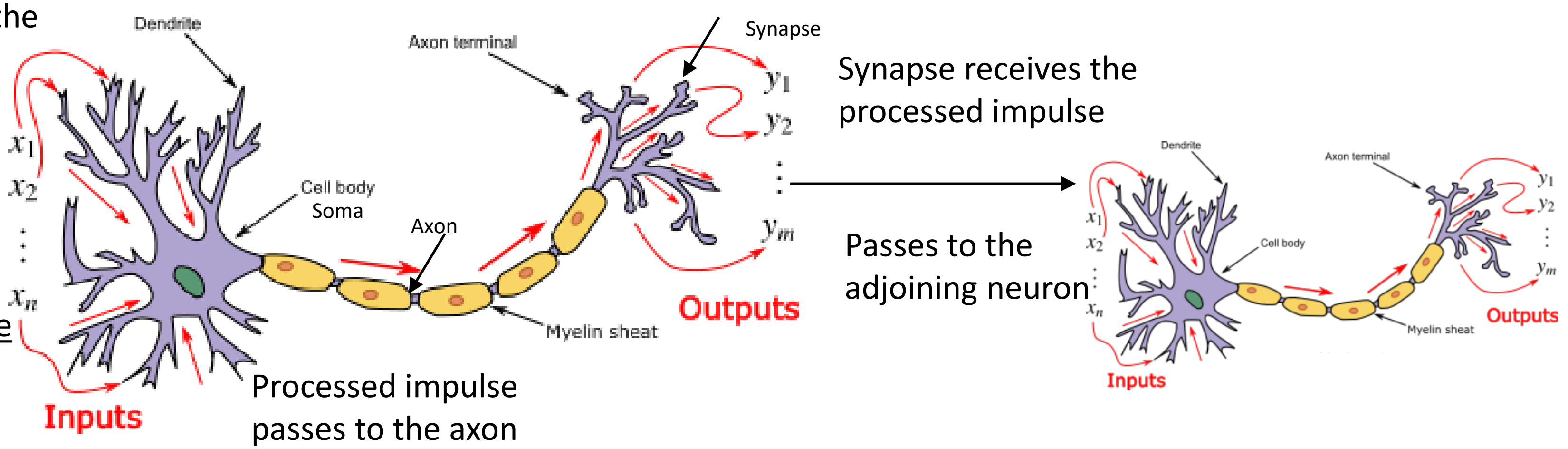
Neuron

- Human brain has a net of neurons (neural networks) – between 14 and 16 billion neurons in cerebral cortex
- Neurons are responsible for transmitting and processing information that we receive from our senses
- Dendrites: receive the information
- Synapses: transmit the processed information
- Soma (cell body) processes the impulses from dendrites and sends the processed impulse to the axon
- Axon: conducting structure through which the processed information is passed

Biological Neural Networks

Receives the impulse from the synapse of another neuron

The impulse goes to the nucleus in the cell body (soma)
<u>Impulse gets processed here</u>



Processed impulse passes to the axon

Synapse receives the processed impulse

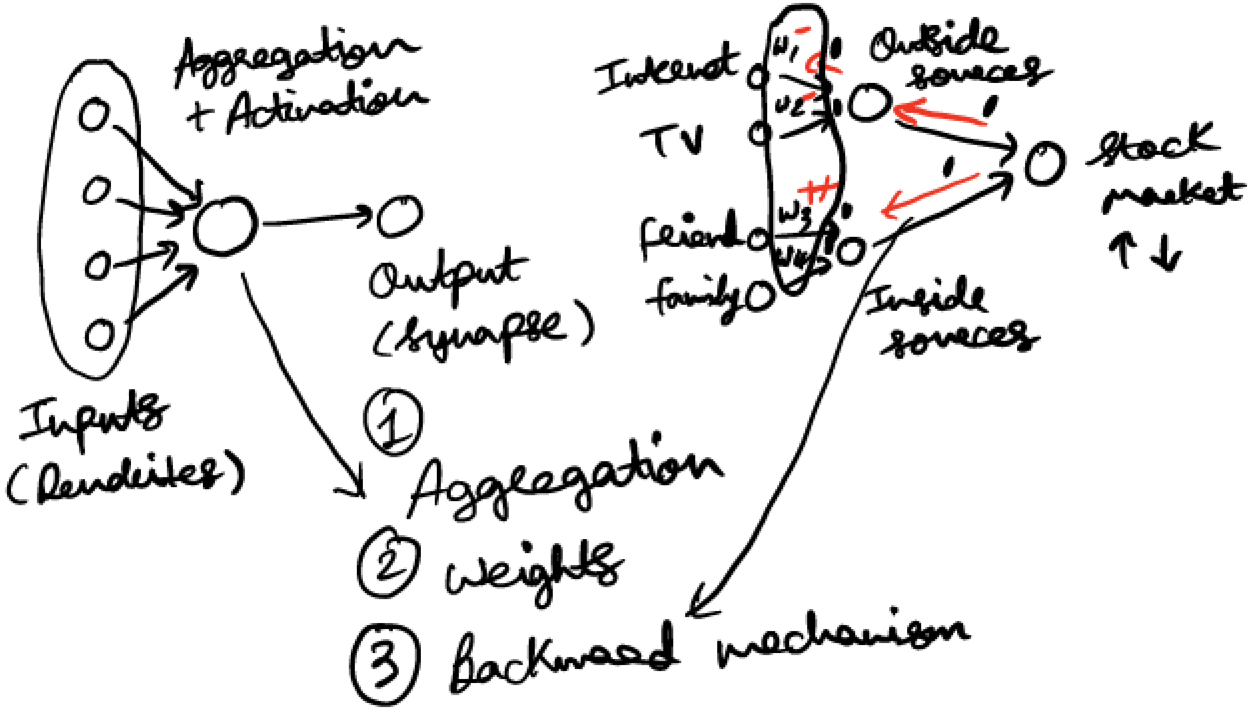
Passes to the adjoining neuron

Neuron

- Some impulses are more important than others and can trigger a neuron to fire easier
- In reality, there is no physical connection between neurons – chemicals are used to communicate the signals (impulses) from synapses to dendrites among neurons
- One neuron is connected to an adjoining neuron at either the entry or exit point(s)
- The neural networks learn patterns (which neurons to fire and the magnitude of signals) and create memories in our brain

Biological Neural Networks: An Example

I am providing the snapshot from the lecture as a placeholder.
Please go through the recorded lecture to understand the details.



Linear Regression: Predict House Prices

Response Variable

House Price

Predictor Variables

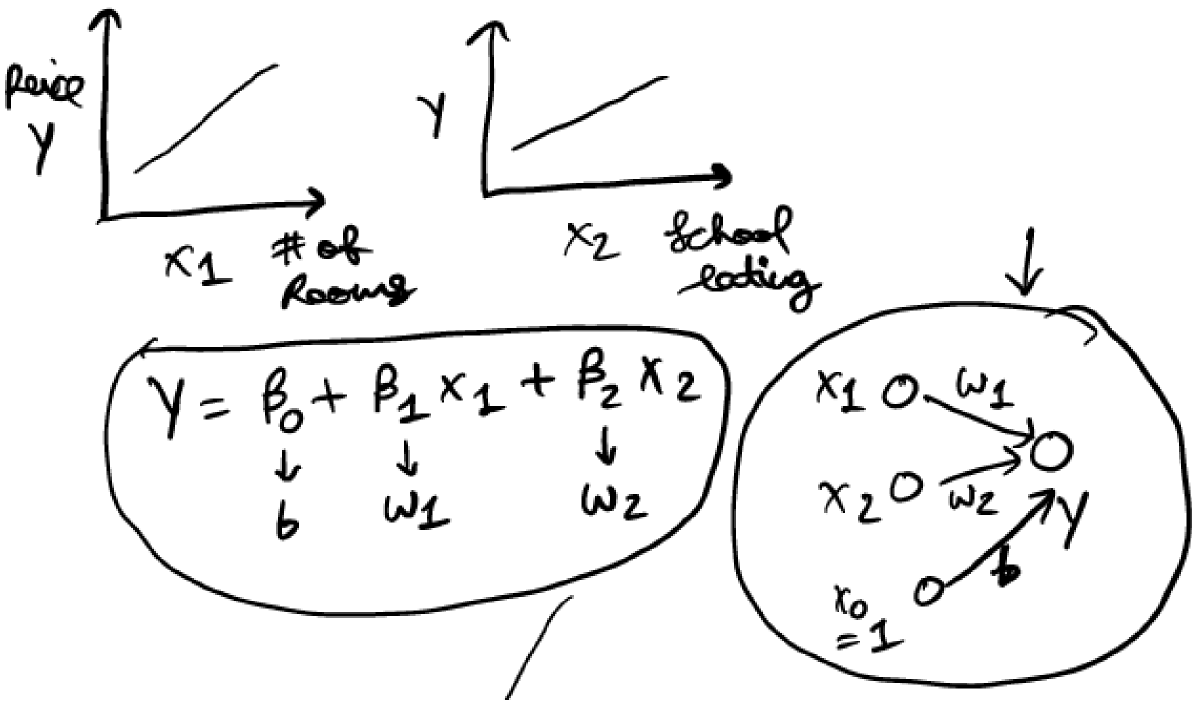
School Ratings

integer values (1-10)

Number of Rooms

integer values (1-10)

New Representation



I am providing the snapshot from the lecture as a placeholder.
Please go through the recorded lecture to understand the details.

Add New Predictors: Mimic a Neural Network

Response Variable

House Price

Predictor Variables

Crime Rate

value between 0 and 1

School Ratings

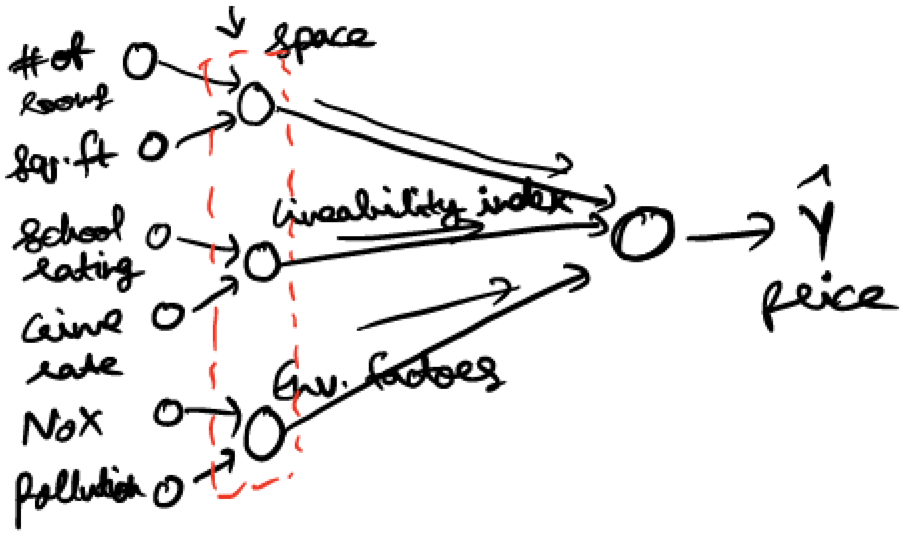
integer values (1-10)

Number of Rooms

integer values (1-10)

Nitric Oxides Concentration

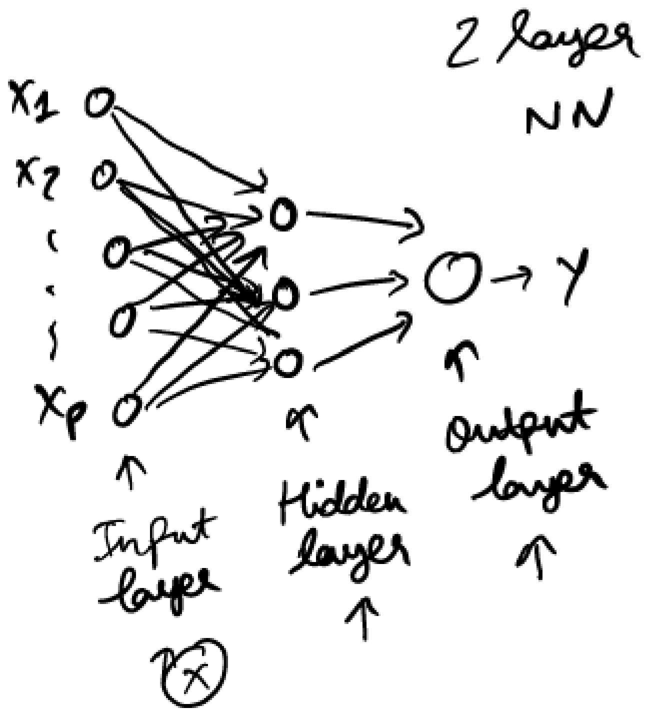
value between 0 and 1



I am providing the snapshot from the lecture as a placeholder.
Please go through the recorded lecture to understand the details.

Generalized Representation of Neural Networks

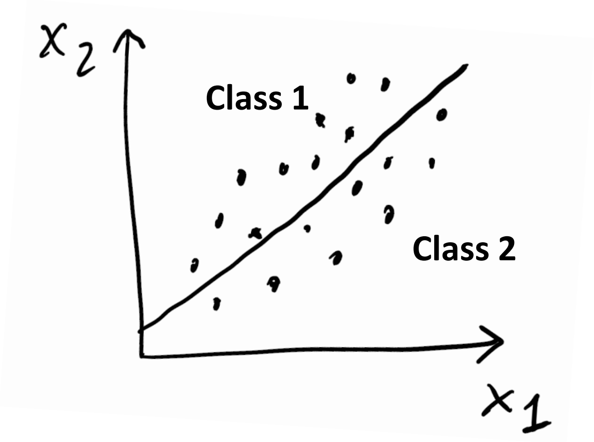
I am providing the snapshot from the lecture as a placeholder.
Please go through the recorded lecture to understand the details.



Also, called a Feedforward Neural Network

Classification

- The objective in classification is to predict a qualitative (categorical or nominal) response outcome, given a set of predictor variable values



- To construct a classifier, we partition the sample space of possible values of X into **non-overlapping** regions
- We give each region a predicted class

- Can we use Linear Regression to create a two-class classifier?
  - If there are only 2 possible outcomes for the response variable, we can assign them as 0-1 and use OLS regression to fit a linear model and obtain a classification boundary (for e.g., class 1 if $\hat{y} > 0.5$)
  - The fitted OLS model: $E(Y|x) = P(Y = 1|x)$ = $\beta_0 + \beta_1 x$
  - The problem here is that unless β_1= 0, the estimates of $P(Y = 1|x)$ will be more than 1 for some values of x

Logistic Regression

- To counter the issue of some predicted probabilities going outside the range of [0,1] when using a linear function, we use the logistic function:

$$p(x) = \frac{\mathrm{e}^{(\beta_0+\beta_1 x\ )}}{1 + \mathrm{e}^{(\beta_0+\beta_1 x\ )}}$$

- For any values of the coefficients, positive, negative or 0, $p(x)$ will belong to (0,1)

Please go through the recorded lecture to understand the details.

Estimating the Parameters

- How to estimate w and b?
- Loss function: measures how well we predict $\hat{y}$ with respect to the ground truth label y <u>for each data point</u> in the training set
- Cost function: measures how well the parameters w and b are doing on the <u>entire training set</u> (with respect to the model fit)

Please go through the recorded lecture to understand the details.

Gradient Descent

- How to train the model?

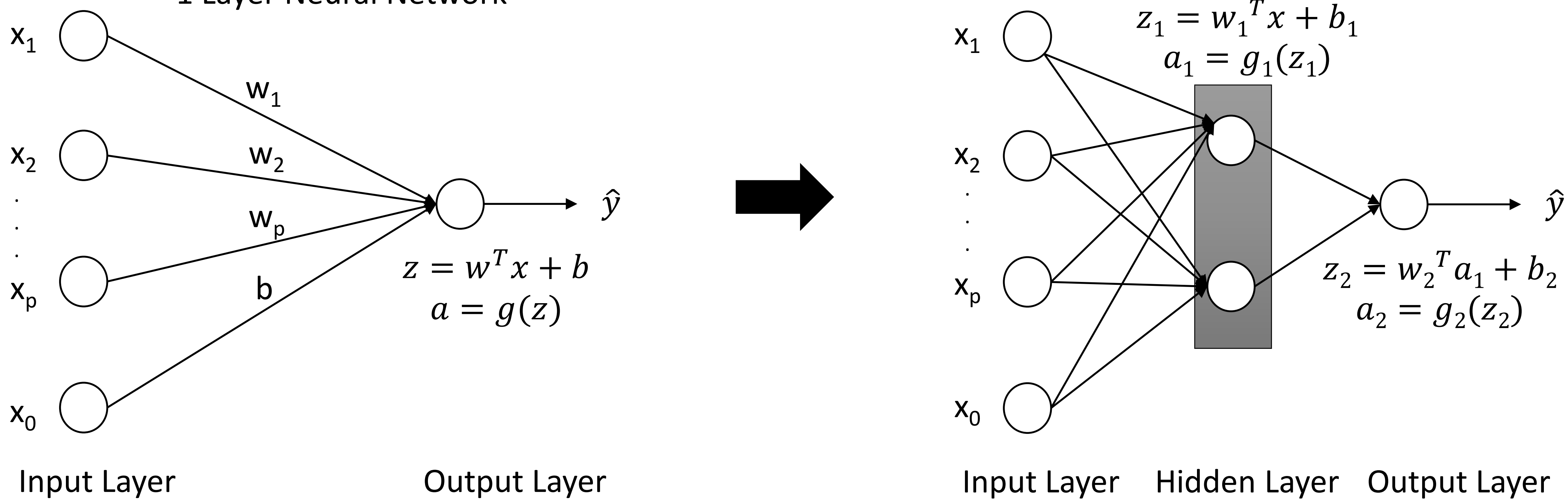
Please go through the recorded lecture to understand the details.

Forward Pass and Backward Pass

Please go through the recorded lecture to understand the details.

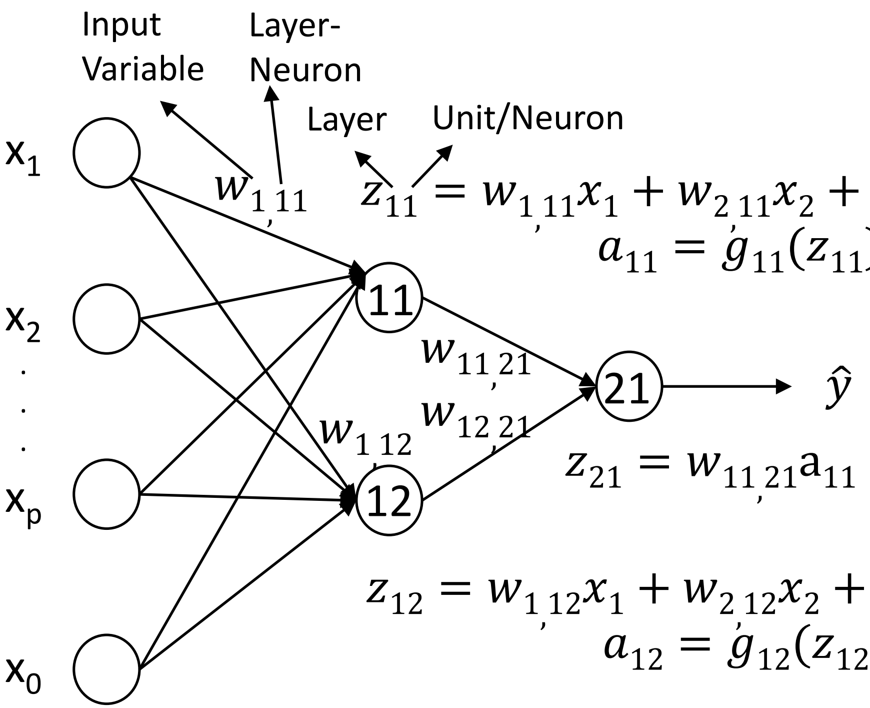
Representation of Neural Networks

Logistic Regression as a
1 Layer Neural Network



Please go through the recorded lecture to understand the details.

Linear Activation Function



$$z_{11} = w_{1,11}x_1 + w_{2,11}x_2 + w_{p,11}x_p + b_1$$
$$a_{11} = g_{11}(z_{11})$$

$$z_{21} = w_{11,21}a_{11} + w_{12,21}a_{12} + b_2$$

$$z_{12} = w_{1,12}x_1 + w_{2,12}x_2 + w_{p,12}x_p + b_1$$
$$a_{12} = g_{12}(z_{12})$$

Please go through the recorded lecture to understand the details.

Regression Problem: Using a Neural Network

Response Variable

House Price

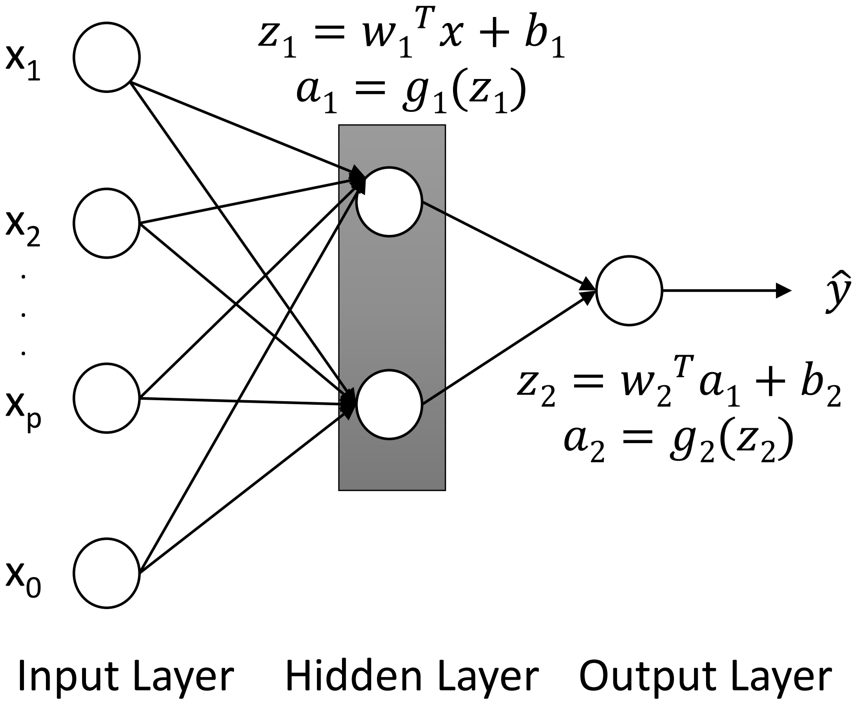
Predictor Variables

Crime Rate
value between 0 and 1

School Ratings
integer values (1-10)

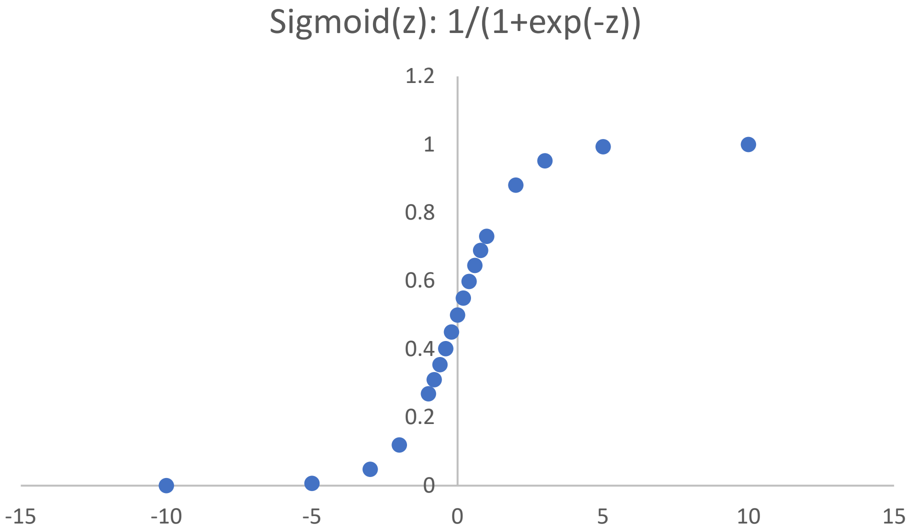
Number of Rooms
integer values (1-10)

Nitric Oxides Concentration
value between 0 and 1

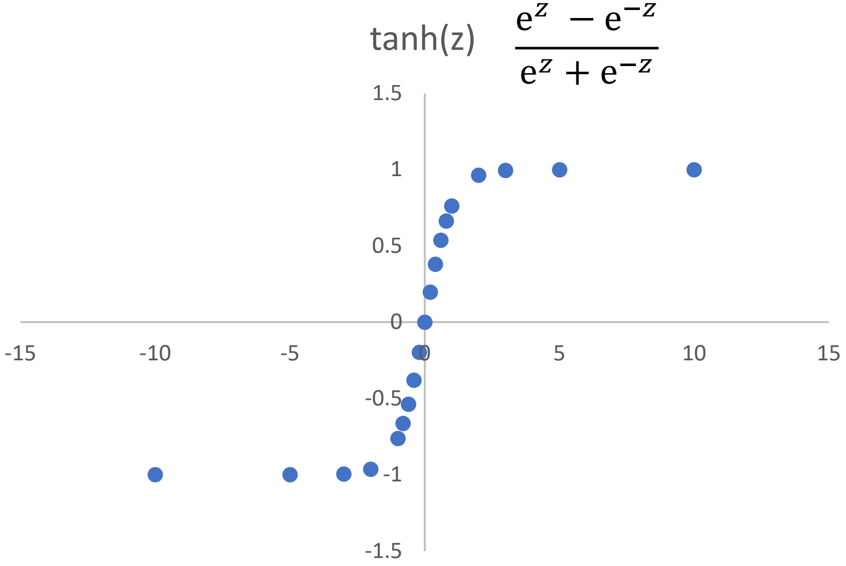


Other Activation Functions

z	1/(1+exp(-z))
-0.2	0.450166003
-0.4	0.40131234
-0.6	0.354343694
-0.8	0.310025519
-1	0.268941421
-2	0.119202922
-3	0.047425873
-5	0.006692851
-10	4.53979E-05
0	0.5
0.2	0.549833997
0.4	0.59868766
0.6	0.645656306
0.8	0.689974481
1	0.731058579
2	0.880797078
3	0.952574127
5	0.993307149
10	0.999954602



z	tanh(z)
-0.2	-0.19738
-0.4	-0.37995
-0.6	-0.53705
-0.8	-0.66404
-1	-0.76159
-2	-0.96403
-3	-0.99505
-5	-0.99991
-10	-1
0	0
0.2	0.197375
0.4	0.379949
0.6	0.53705
0.8	0.664037
1	0.761594
2	0.964028
3	0.995055
5	0.999909
10	1



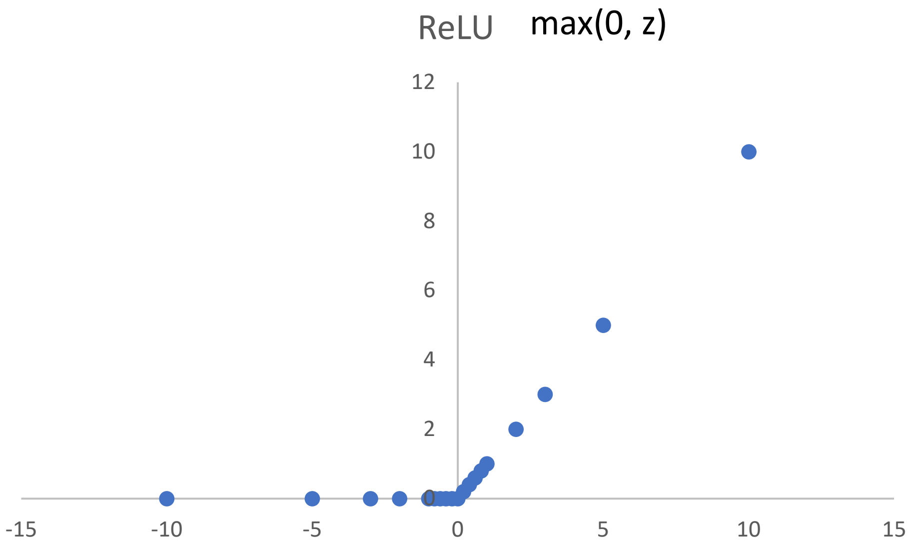
$$\frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$$

Please go through the recorded lecture to understand the details.

Other Activation Functions

z	ReLU
-0.2	0
-0.4	0
-0.6	0
-0.8	0
-1	0
-2	0
-3	0
-5	0
-10	0
0	0
0.2	0.2
0.4	0.4
0.6	0.6
0.8	0.8
1	1
2	2
3	3
5	5
10	10

max(0, z)



Please go through the recorded lecture to understand the details.

Parameters and Hyperparameters

- Hyperparameter is a parameter whose value is set before the learning process begins
- Whereas the values of the other parameters are derived upon learning

Please go through the recorded lecture to understand the details.
(List of parameters and hyperparameters in Neural Networks)

Neural Networks (in Python)

Import the module

```
from sklearn.neural_network import MLPClassifier
```

For Regression - MLPRegressor

Create the model

```
model =  MLPClassifier() #a list of parameters that can be passed
```

Optimizer: Adam
(Adaptive Moment Estimation)

Fit the model (on training data)

```
model.fit(x_train, y_train)
```

Predict y_hat values for the test data

```
y_hat = model.predict(x_test)
```

Calculate the accuracy: First import the module:

```
from sklearn.metrics import confusion_matrix, accuracy_score
confusion_matrix(y_test, y_hat)
accuracy_score(y_test, y_hat)
```

Result from the final code execution from recorded lecture (#19): NN_Class_Example_2 (regression)

```
y_pred_NN = grid_search.predict(x_training_set)
```

```
mse_grid = mean_squared_error(y_training_set, y_pred_NN)
```

```
mse_grid
```

```
6.726265279232006
```