Workshop 6 – Deep Learning - BackProp

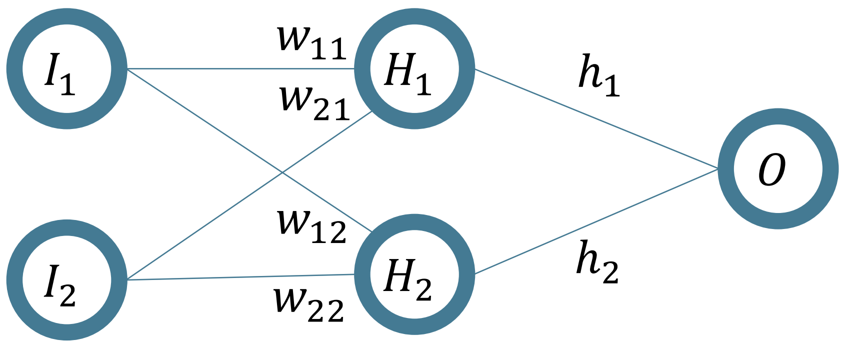
Advanced Analytics and Applications [AAA]

Information Systems for Sustainable Society (is3) | WiSo Faculty | Univ.-Prof. Dr. Wolfgang Ketter | 5/15/23

I Calculation

II Programming

Question 2: Calculation BackProp



Assume we have the following neural network setup.

- Learning rate $\eta = 0.05$
- Linear activation function
- One Data Point:

  $I_1 = 2 \quad I_2 = 3 \quad O = 1$

- Random Initial weights:

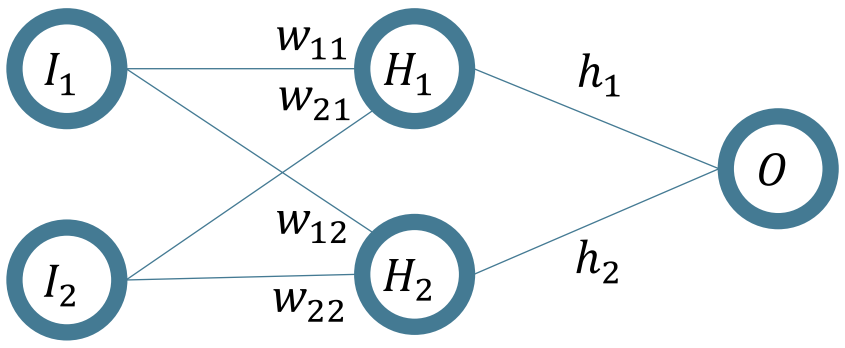
  $w_{11} = 0.11 \quad h_1 = 0.14$

  $w_{21} = 0.21 \quad h_2 = 0.15$

  $w_{12} = 0.12$

  $w_{22} = 0.08$

Question 2: Calculation BackProp



$$\hat{O} = \begin{bmatrix} 2 & 3 \end{bmatrix} * \begin{bmatrix} 0.11 & 0.12 \\ 0.21 & 0.08 \end{bmatrix} * \begin{bmatrix} 0.14 \\ 0.15 \end{bmatrix} = 0.191$$

a) Calculate the prediction using this configuration.

- Learning rate $\eta = 0.05$
- Linear activation function
- One Data Point:

  $I_1 = 2$ $I_2 = 3$ $O = 1$

- Random Initial weights:

  $w_{11} = 0.11$ $h_1 = 0.14$

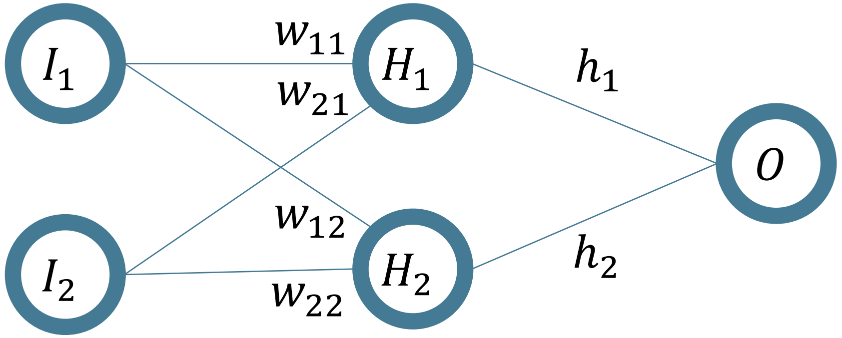
  $w_{21} = 0.21$ $h_2 = 0.15$

  $w_{12} = 0.12$

  $w_{22} = 0.08$

Universität zu Köln

Question 2: Calculation BackProp



$$L = \frac{1}{1}(0.191-1)^2 = 0.655$$

b) Calculate the error using MSE loss function

- Learning rate $\eta = 0.05$
- Linear activation function
- One Data Point:

$I_1 = 2 \quad I_2 = 3 \quad O = 1$

- Random Initial weights:

$w_{11} = 0.11 \quad h_1 = 0.14$

$w_{21} = 0.21 \quad h_2 = 0.15$

$w_{12} = 0.12$

$w_{22} = 0.08$

Question 2: Calculation BackProp

c) Calculate Derivative of Loss with respect to h_1

- Learning rate $\eta = 0.05$
- Linear activation function
- One Data Point:

$I_1 = 2 \quad I_2 = 3 \quad O = 1$

- Random Initial weights:

$w_{11} = 0.11 \quad h_1 = 0.14$

$w_{21} = 0.21 \quad h_2 = 0.15$

$w_{12} = 0.12$

$w_{22} = 0.08$

$$\frac{\partial L}{\partial h_1} = \frac{\partial L}{\partial \hat{O}} * \frac{\partial \hat{O}}{\partial h_1}$$

$$= \frac{\partial((\hat{O} - O)^2)}{\partial \hat{O}} * \frac{\partial[(I_1 * w_{11} + I_2 w_{21})h_1 + (I_1 * w_{12} + I_2 w_{22})h_2)}{\partial h_1}$$

$$= \frac{\partial((\hat{O} - O))}{\partial \hat{O}} * 2 * (\hat{O} - O) * (I_1 * w_{11} + I_2 w_{21})$$

$$= 2 * (\hat{O} - O) * (I_1 * w_{11} + I_2 w_{21})$$

Similarly for $\frac{\partial L}{\partial h_2} = 2 * (\hat{O} - O) * (I_1 * w_{12} + I_2 w_{22})$

Question 2: Calculation BackProp

c) Calculate Derivative of Loss with respect to w_{11}

$$\frac{\partial L}{\partial w_{11}} = \frac{\partial L}{\partial \hat{O}} * \frac{\partial \hat{O}}{\partial H_1} * \frac{\partial H_1}{\partial w_{11}}$$

$$= \frac{\partial((\hat{O} - O)^2)}{\partial \hat{O}} * \frac{\partial[(H_1)h_1 + (H_2)h_2)]}{\partial H_1} * \frac{\partial H_1}{\partial w_{11}}$$

$$\boxed{(I_1 * w_{11} + I_2 w_{21})}$$

$$= \frac{\partial((\hat{O} - O))}{\partial \hat{O}} * 2 * (\hat{O} - O) * (h_1) * (I_1)$$

$$= 2 * (\hat{O} - O) * (h_1) * (I_1)$$

Similarly for $\frac{\partial L}{\partial w_{12}} = 2 * (\hat{O} - O) * (h_2) * (I_1)$

$$\frac{\partial L}{\partial w_{21}} = 2 * (\hat{O} - O) * (h_1) * (I_2)$$

$$\frac{\partial L}{\partial w_{22}} = 2 * (\hat{O} - O) * (h_2) * (I_2)$$

- Learning rate $\eta = 0.05$
- Linear activation function
- One Data Point:

  $I_1 = 2$ $I_2 = 3$ $O = 1$

- Random Initial weights:

  $w_{11} = 0.11$ $h_1 = 0.14$

  $w_{21} = 0.21$ $h_2 = 0.15$

  $w_{12} = 0.12$

  $w_{22} = 0.08$

Universität zu Köln

Question 2: Calculation BackProp

$$w_{11}^{k+1} = w_{11}^{k} - \eta * 2 * (\hat{O} - O) * (h_1) * (I_1)$$

$$w_{12}^{k+1} = w_{12}^{k} - \eta * 2 * (\hat{O} - O) * (h_2) * (I_1)$$

$$w_{21}^{k+1} = w_{21}^{k} - \eta * 2 * (\hat{O} - O) * (h_1) * (I_2)$$

$$w_{22}^{k+1} = w_{22}^{k} - \eta * 2 * (\hat{O} - O) * (h_2) * (I_2)$$

$$h_1^{k+1} = h_1^{k} - \eta * 2 * (\hat{O} - O) * (I_1 * w_{11} + I_2 w_{21})$$

$$h_2^{k+1} = h_2^{k} - \eta * 2 * (\hat{O} - O) * (I_1 * w_{12} + I_2 w_{22})$$

Updating weights using derivates

- Learning rate $\eta = 0.05$
- Linear activation function
- One Data Point:

  $I_1 = 2$ $I_2 = 3$ $O = 1$

- Random Initial weights:

  $w_{11} = 0.11$ $h_1 = 0.14$

  $w_{21} = 0.21$ $h_2 = 0.15$

  $w_{12} = 0.12$

  $w_{22} = 0.08$

Question 2: Calculation BackProp

$$w_{11}^1 = w_{11}^0 - \eta * 2 * (\hat{O} - O) * (h_1) * (I_1)$$
$$= 0.11 - 0.05 * 2 * (0.191 - 1) * 0.14 * 2$$
$$\cong 0.13$$

$$w_{12}^1 = 0.14$$

$$w_{21}^1 = 0.23$$

$$w_{22}^1 = 0.10$$

$$h_1^1 = 0.21$$

$$h_2^1 = 0.19$$

d) Updating weights using derivates with real values

- Learning rate $\eta = 0.05$
- Linear activation function
- One Data Point:

  $I_1 = 2$ $I_2 = 3$ $O = 1$

- Random Initial weights:

  $w_{11} = 0.11$ $h_1 = 0.14$

  $w_{21} = 0.21$ $h_2 = 0.15$

  $w_{12} = 0.12$

  $w_{22} = 0.08$

Question 2: Calculation BackProp

e) Calculate prediction using updated weights

- Learning rate $\eta = 0.05$
- Linear activation function
- One Data Point:

  $I_1 = 2 \quad I_2 = 3 \quad O = 1$

- Random Initial weights:

  $w_{11} = 0.11 \quad h_1 = 0.14$

  $w_{21} = 0.21 \quad h_2 = 0.15$

  $w_{12} = 0.12$

  $w_{22} = 0.08$

Results in updated prediction:

$$\hat{O} = \begin{bmatrix} 2 & 3 \end{bmatrix} * \begin{bmatrix} 0.13 & 0.14 \\ 0.23 & 0.10 \end{bmatrix} * \begin{bmatrix} 0.21 \\ 0.17 \end{bmatrix} = 0.2981$$

We got closer to the real O – that‘s how backprop works in a nutshell 🙂

Universität zu Köln

I Calculation

II Programming

Outline

1 Classifying newswires

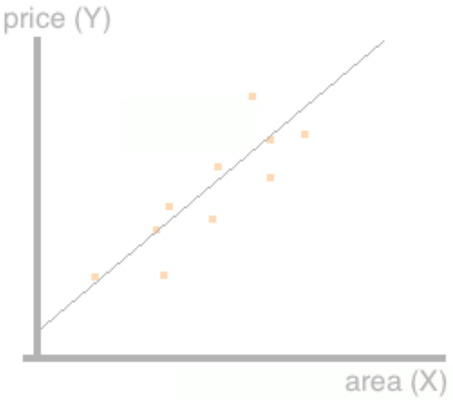
2 Regression with Keras

https://towardsdatascience.com/text-classification-in-keras-part-1-a-simple-reuters-news-classifier-9558d34d01d3
https://medium.com/rocknnull/playing-with-machine-learning-a-practical-example-using-keras-tensorflow-790375cd1abb

Outline

1 Classifying newswires

2 Regression with Keras



https://towardsdatascience.com/text-classification-in-keras-part-1-a-simple-reuters-news-classifier-9558d34d01d3
https://medium.com/rocknnull/playing-with-machine-learning-a-practical-example-using-keras-tensorflow-790375cd1abb

Universität zu Köln

The dataset:Reuters-21578

https://towardsdatascience.com/text-classification-in-keras-part-1-a-simple-reuters-news-classifier-9558d34d01d3
https://medium.com/rocknnull/playing-with-machine-learning-a-practical-example-using-keras-tensorflow-790375cd1abb

Universität zu Köln

Fact sheet: Reuters-21578

11228 newswires

Over 46 topics

https://towardsdatascience.com/text-classification-in-keras-part-1-a-simple-reuters-news-classifier-9558d34d01d3
https://medium.com/rocknnull/playing-with-machine-learning-a-practical-example-using-keras-tensorflow-790375cd1abb

Fact sheet: Reuters-21578

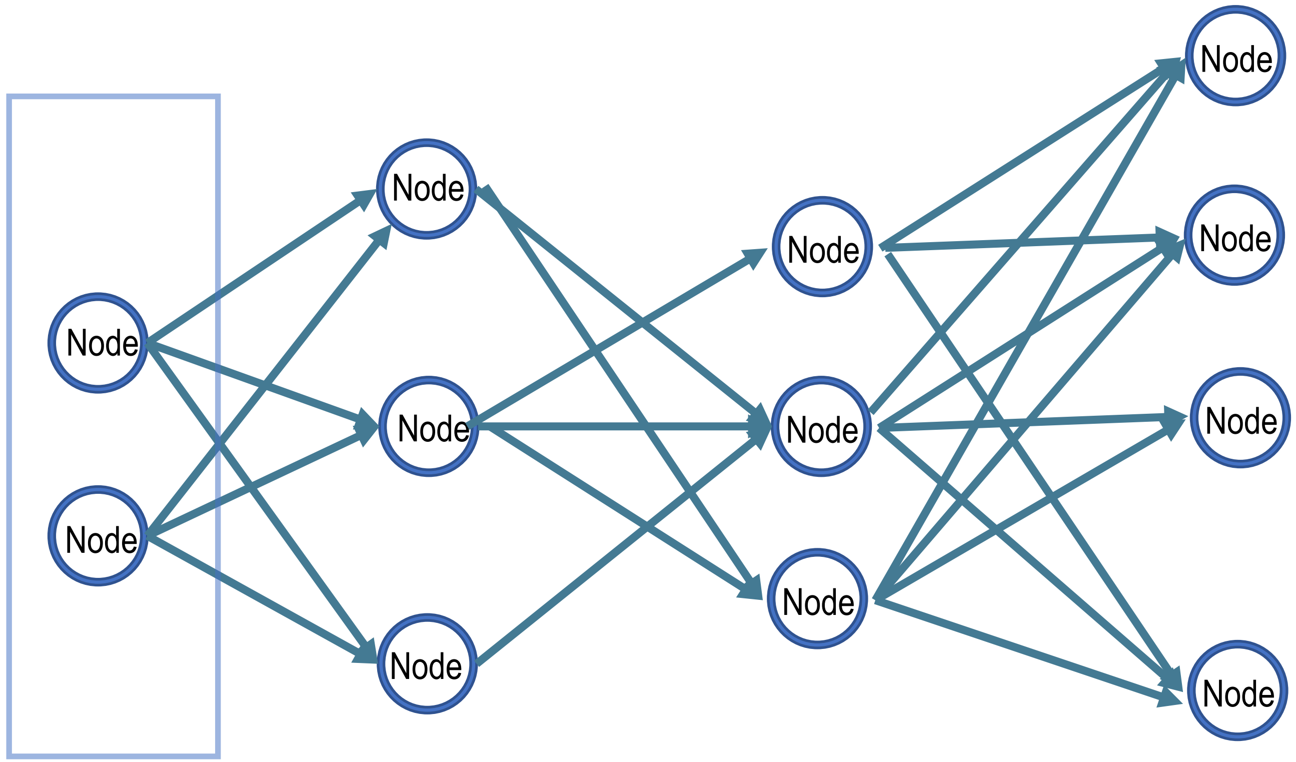
1. Toy dataset in **keras**.
2. Each newswire is encoded as a list of **word indexes**.
3. For convenience, words are indexed by overall **frequency** in the dataset, so that **for instance** the **integer "3" encodes the 3rd most frequent** word in the data.

https://towardsdatascience.com/text-classification-in-keras-part-1-a-simple-reuters-news-classifier-9558d34d01d3
https://medium.com/rocknnull/playing-with-machine-learning-a-practical-example-using-keras-tensorflow-790375cd1abb

Our objective: Train a deep learning network that can classify unseen documents

https://towardsdatascience.com/text-classification-in-keras-part-1-a-simple-reuters-news-classifier-9558d34d01d3
https://medium.com/rocknnull/playing-with-machine-learning-a-practical-example-using-keras-tensorflow-790375cd1abb

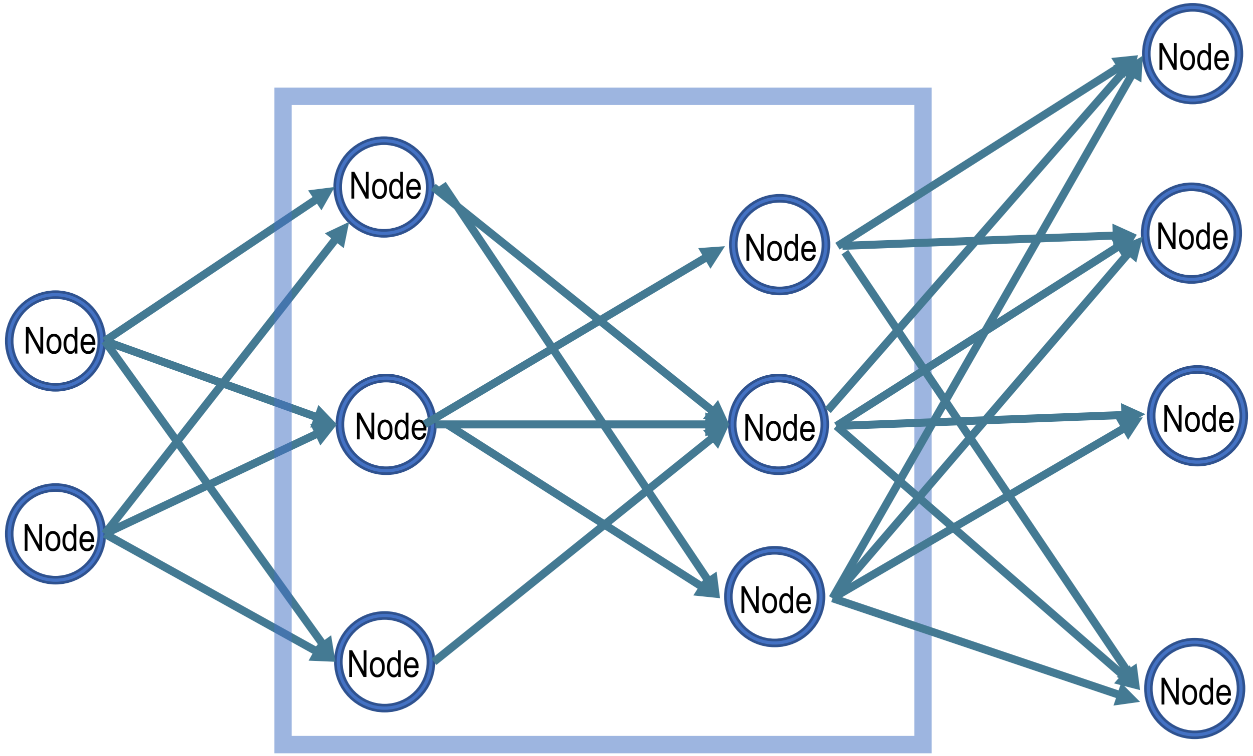
Our objective: Train a deep learning network that can classify unseen documents



Input Layer with 10000 Input Nodes

https://towardsdatascience.com/text-classification-in-keras-part-1-a-simple-reuters-news-classifier-9558d34d01d3
https://medium.com/rocknnull/playing-with-machine-learning-a-practical-example-using-keras-tensorflow-790375cd1abb

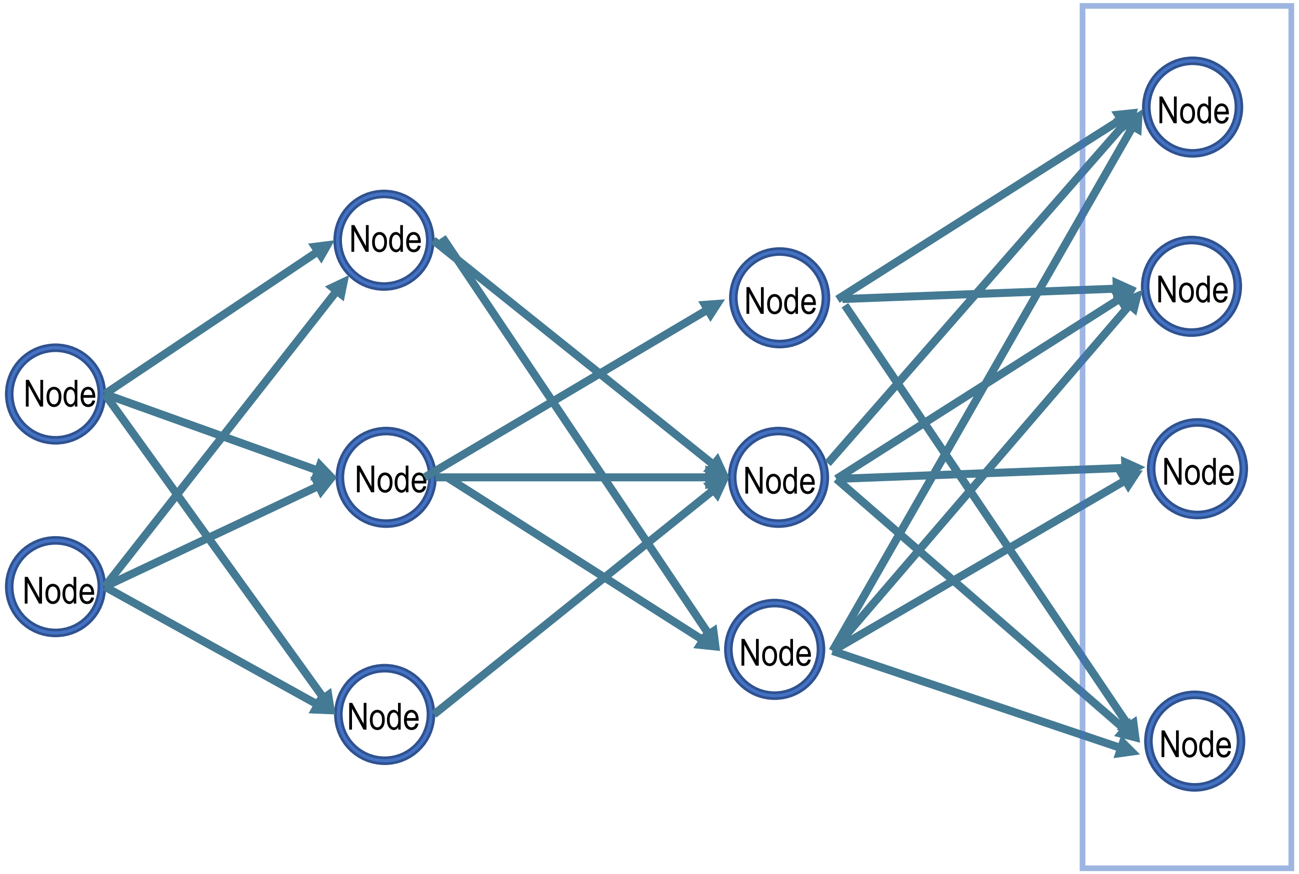
Our objective: Train a deep learning network that can classify unseen documents



64 Nodes, 2 Hidden Layers,
Activation Function: **ReLU**

https://towardsdatascience.com/text-classification-in-keras-part-1-a-simple-reuters-news-classifier-9558d34d01d3
https://medium.com/rocknnull/playing-with-machine-learning-a-practical-example-using-keras-tensorflow-790375cd1abb

Our objective: Train a deep learning network that can classify unseen documents



46 Nodes, 1 Output Layer,
Activation Function: **SoftMAX**

https://towardsdatascience.com/text-classification-in-keras-part-1-a-simple-reuters-news-classifier-9558d34d01d3
https://medium.com/rocknnull/playing-with-machine-learning-a-practical-example-using-keras-tensorflow-790375cd1abb

Outline

1 Classifying newswires

2 Regression with Keras

https://towardsdatascience.com/text-classification-in-keras-part-1-a-simple-reuters-news-classifier-9558d34d01d3
https://medium.com/rocknnull/playing-with-machine-learning-a-practical-example-using-keras-tensorflow-790375cd1abb

Universität zu Köln

The boston housing price data set

Price of House?

https://towardsdatascience.com/linear-regression-on-boston-housing-dataset-f409b7e4a155

The boston housing price data set

1. Sample contains 404 training and 102 test samples.
2. 13 attributes (independent variables) – i.e., attributes of the houses at different location
3. The target value are the median values of the houses at a location (in k $)

https://towardsdatascience.com/linear-regression-on-boston-housing-dataset-f409b7e4a155

The boston housing price data set

```
Variables in order:
CRIM     per capita crime rate by town
ZN       proportion of residential land zoned for lots over 25,000 sq.ft.
INDUS    proportion of non-retail business acres per town
CHAS     Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
NOX      nitric oxides concentration (parts per 10 million)
RM       average number of rooms per dwelling
AGE      proportion of owner-occupied units built prior to 1940
DIS      weighted distances to five Boston employment centres
RAD      index of accessibility to radial highways
TAX      full-value property-tax rate per $10,000
PTRATIO  pupil-teacher ratio by town
B        1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
LSTAT    % lower status of the population
MEDV     Median value of owner-occupied homes in $1000's
```


https://towardsdatascience.com/linear-regression-on-boston-housing-dataset-f409b7e4a155

The boston housing price data set

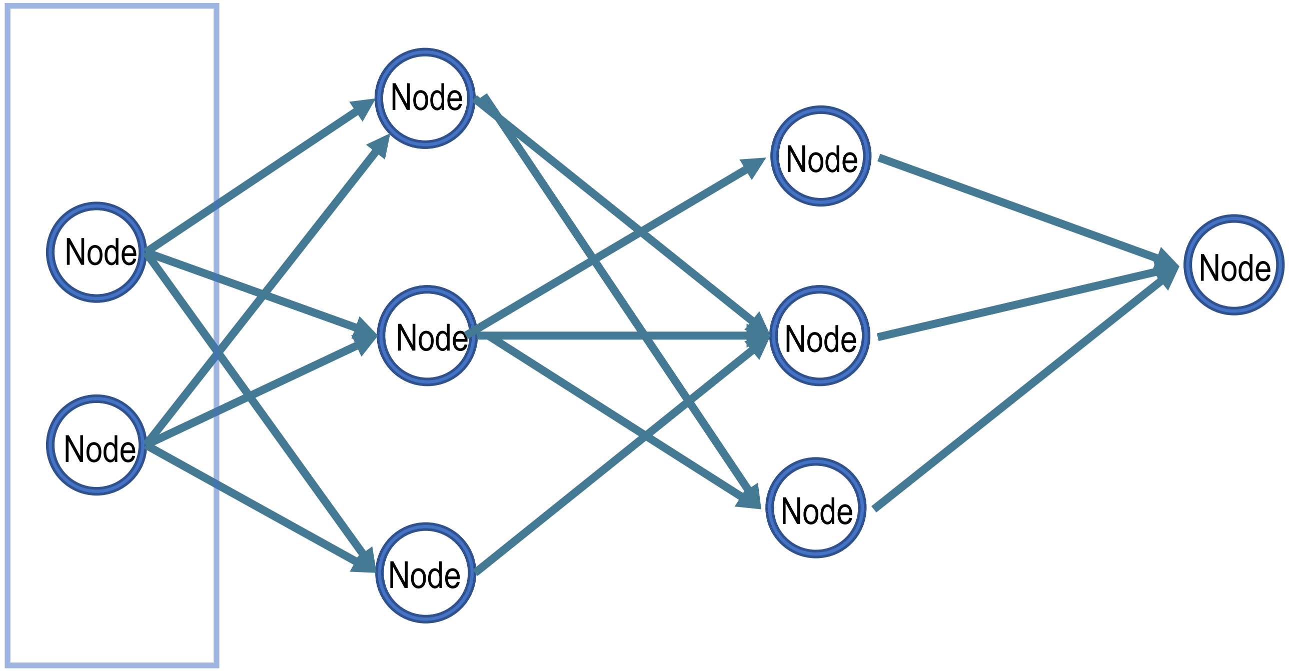
```
Variables in order:
CRIM     per capita crime rate by town
ZN       proportion of residential land zoned for lots over 25,000 sq.ft.
INDUS    proportion of non-retail business acres per town
CHAS     Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
NOX      nitric oxides concentration (parts per 10 million)
RM       average number of rooms per dwelling
AGE      proportion of owner-occupied units built prior to 1940
DIS      weighted distances to five Boston employment centres
RAD      index of accessibility to radial highways
TAX      full-value property-tax rate per $10,000
PTRATIO  pupil-teacher ratio by town
B        1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
LSTAT    % lower status of the population
MEDV     Median value of owner-occupied homes in $1000's
```

The attributes have different scales and ranges.

We need to normalize this before training the network.

https://towardsdatascience.com/linear-regression-on-boston-housing-dataset-f409b7e4a155

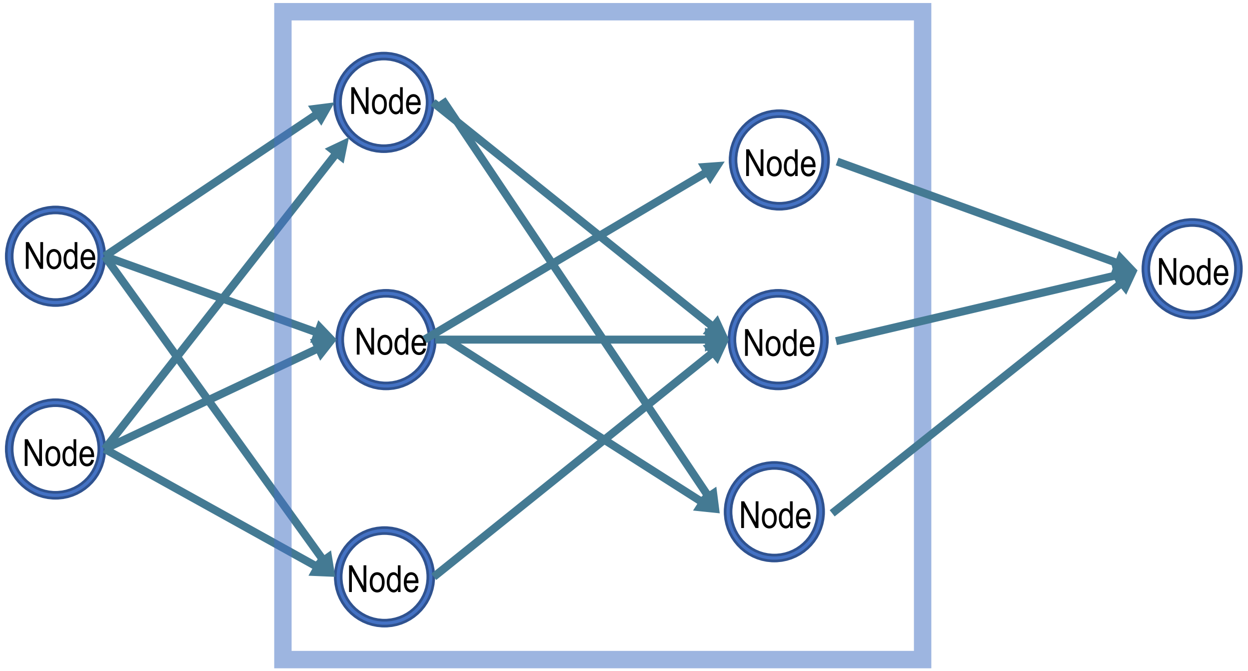
Our objective: Train a deep learning network that can predict house prices



Input Layer with 13 Input Nodes

https://towardsdatascience.com/text-classification-in-keras-part-1-a-simple-reuters-news-classifier-9558d34d01d3
https://medium.com/rocknnull/playing-with-machine-learning-a-practical-example-using-keras-tensorflow-790375cd1abb

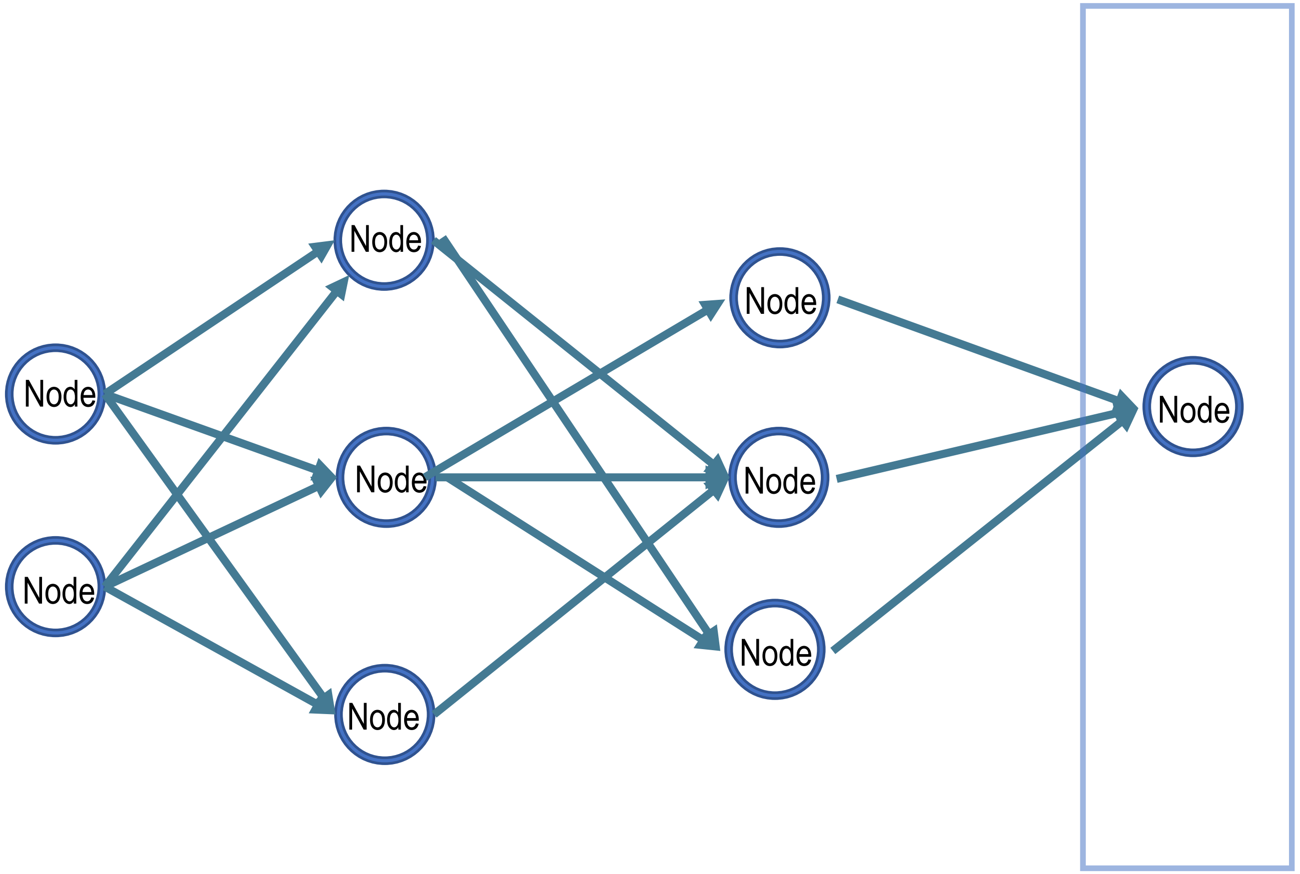
Our objective: Train a deep learning network that can predict house prices



2 Hidden Layers, 64 Nodes in each layer
Activation Function: **ReLU**

https://towardsdatascience.com/text-classification-in-keras-part-1-a-simple-reuters-news-classifier-9558d34d01d3
https://medium.com/rocknnull/playing-with-machine-learning-a-practical-example-using-keras-tensorflow-790375cd1abb

Our objective: Train a deep learning network that can classify unseen documents



Loss: MSE
Metric: MAE

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

$$\mathrm{MAE} = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j|$$

1 Output Node, Real Value Output
Activation Function: ***Linear***

https://towardsdatascience.com/text-classification-in-keras-part-1-a-simple-reuters-news-classifier-9558d34d01d3
https://medium.com/rocknnull/playing-with-machine-learning-a-practical-example-using-keras-tensorflow-790375cd1abb

Contact

For general questions and enquiries on **research**, **teaching**, **job openings** and new **projects** refer to our website at www.is3.uni-koeln.de

For specific enquiries regarding this course contact us by sending an email to the **IS3 teaching** address at is3-teaching@wiso.uni-koeln.de

Universität zu Köln