Introduction to Neural Networks

Econometric Seminar

Neural networks

- inspired by biological neural networks
- it is **framework** for machine learning algorithms
- Types:
  - **Feedforward neural network**
  - Convolutional neural network
  - Recursive neural network

Structure of feedforward neural network

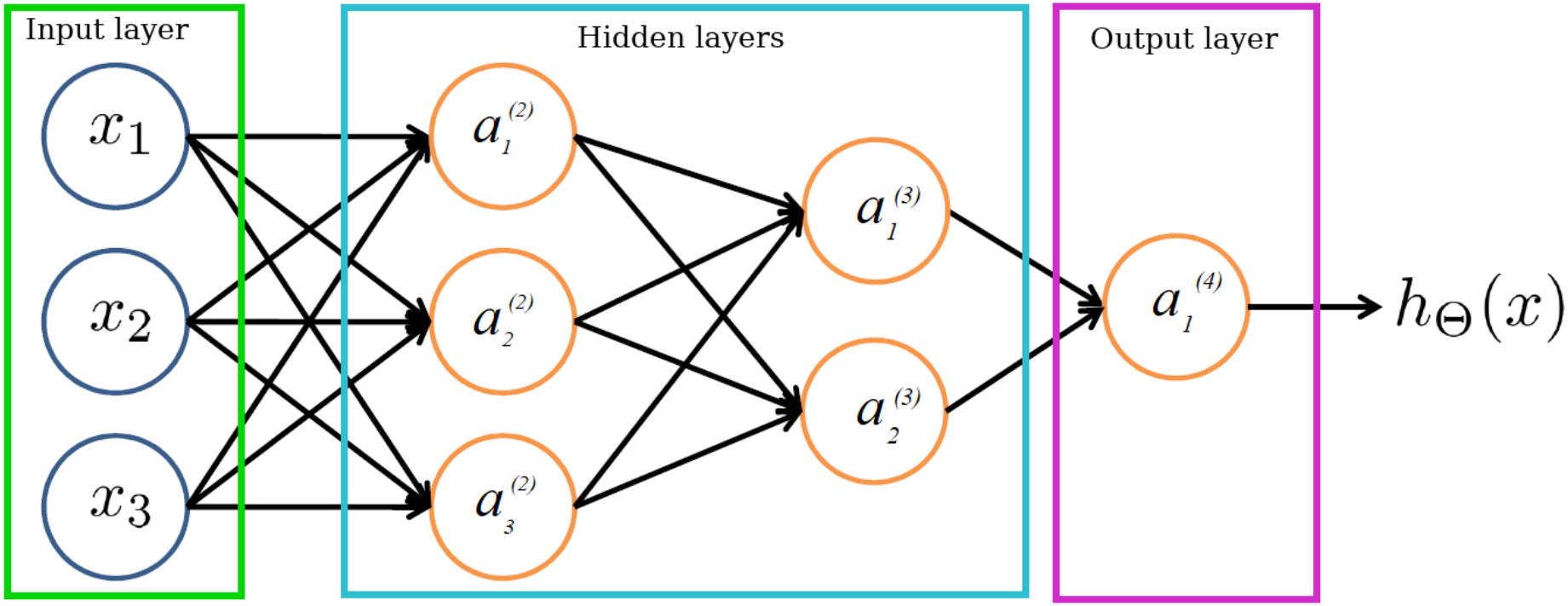

Figure 1: Structure of feedforward neural network

Input layer

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{1}$$

Hidden layers

$$
\begin{aligned}
a_1^{(2)} &= g(w_{11}^{(1)} x_1 + w_{12}^{(1)} x_2 + w_{13}^{(1)} x_3) \\
a_2^{(2)} &= g(w_{21}^{(1)} x_1 + w_{22}^{(1)} x_2 + w_{23}^{(1)} x_3) \\
a_3^{(2)} &= g(w_{31}^{(1)} x_1 + w_{32}^{(1)} x_2 + w_{33}^{(1)} x_3) \\
&\dots \\
a_1^{(3)} &= g(w_{11}^{(2)} a_1^{(2)} + w_{12}^{(2)} a_2^{(2)} + w_{13}^{(2)} a_3^{(2)})
\end{aligned}
$$

- *neurons* $a_j^{(i)}$ receive *impulses* from all neurons in previous layer
- usually some *bias* $w_0^{(i)}$ is added to each layer, then $g(w_0^{(1)} + w_{11}^{(1)} x_1 + \dots)$
- each layer can have different activation function
- each layer can have different number of neurons

Output layer

$$h_W(x) = a^{(4)} = g(w_{11}^{(3)} a_1^{(3)} + w_{12}^{(3)} a_2^{(3)}) \tag{2}$$

- Regression - linear activation function is applied
- Classification - function with output (0,1) is used
- **Output layer** can have more than one neurons, e.g. multiclassification

Activation function

What is function $g()$?

- Sigmoid

$$g(z) = \frac{1}{1 + e^{-z}}$$

- Softmax

$$g(z) = \frac{e^{z}}{\sum_{k=1}^{K} e^{z_k}}$$

- Linear

$$g(z) = z$$

Cost/Loss functions

- Classification

$$J(w) = -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} log(h_w(x^{(i)})) + (1 - y^{(i)}) log/1 - h_w(x^{(i)}))\right]$$

- Regression

$$J(w) = \frac{1}{m}\sum_{i=1}^{m}(h_w(x^{(i)}) - y^{(i)})^2$$

For one output node, otherwise $\sum_{i=1}^{m}\sum_{k=1}^{K}\ldots$, where K is number of output nodes

Estimation

Forward propagation

Information provided by $\mathbf{x}$ propagates to neurons at each hidden layer and produces $\hat{\mathbf{y}} = h_w(\mathbf{x})$.

Backpropagation

Error is "followed" backward through the network in order to compute the gradient. Goal of backpropagation is to iteratively change weights in such a manner that Loss function will be minimized. Commonly used algorithm is gradient descent.

Overfitting (Variance vs Bias)

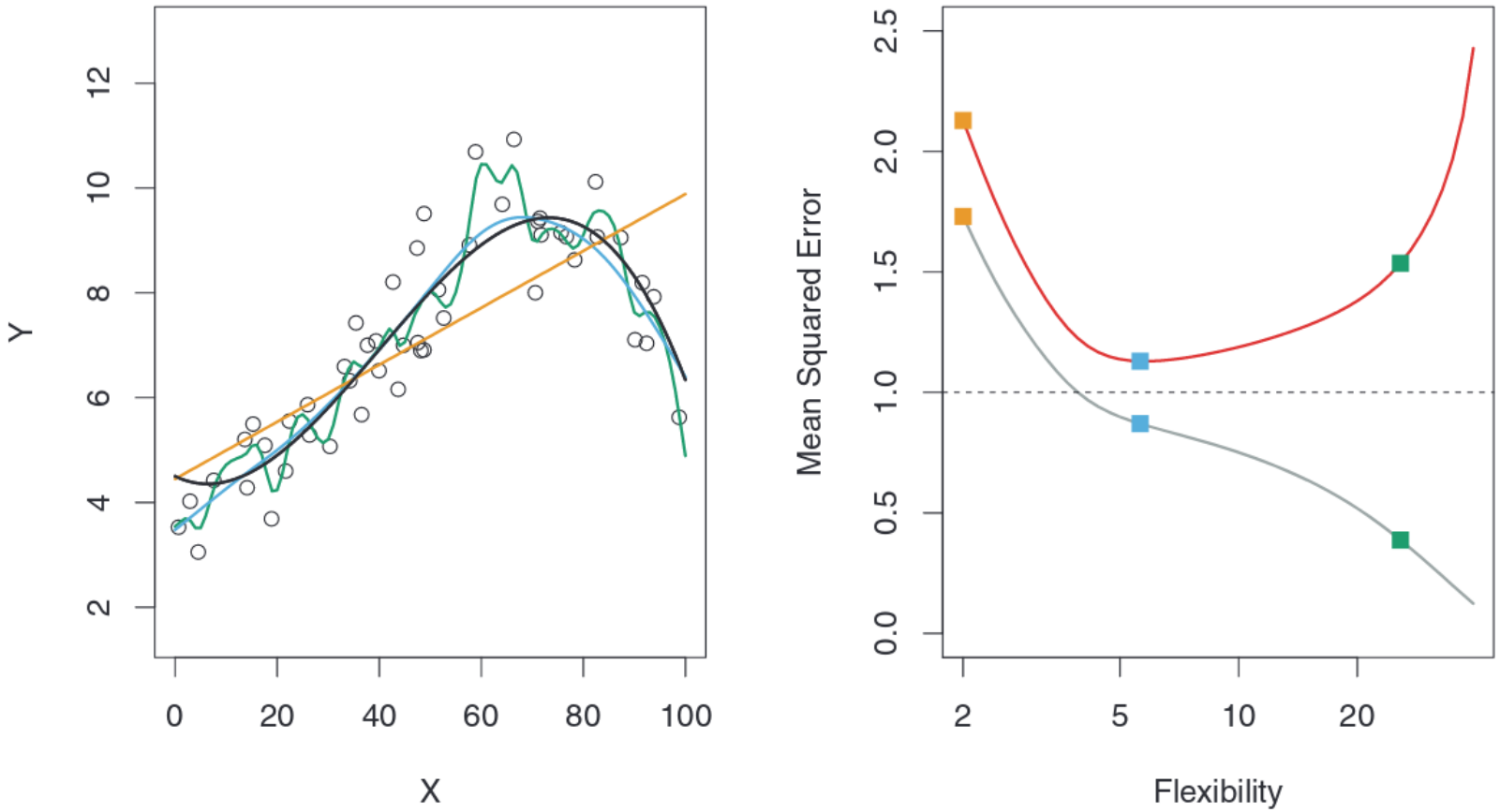

Figure 2: Gray curve (training data), Red curve (test data), Source: Introduction to statistical learning

Overfitting: Regression

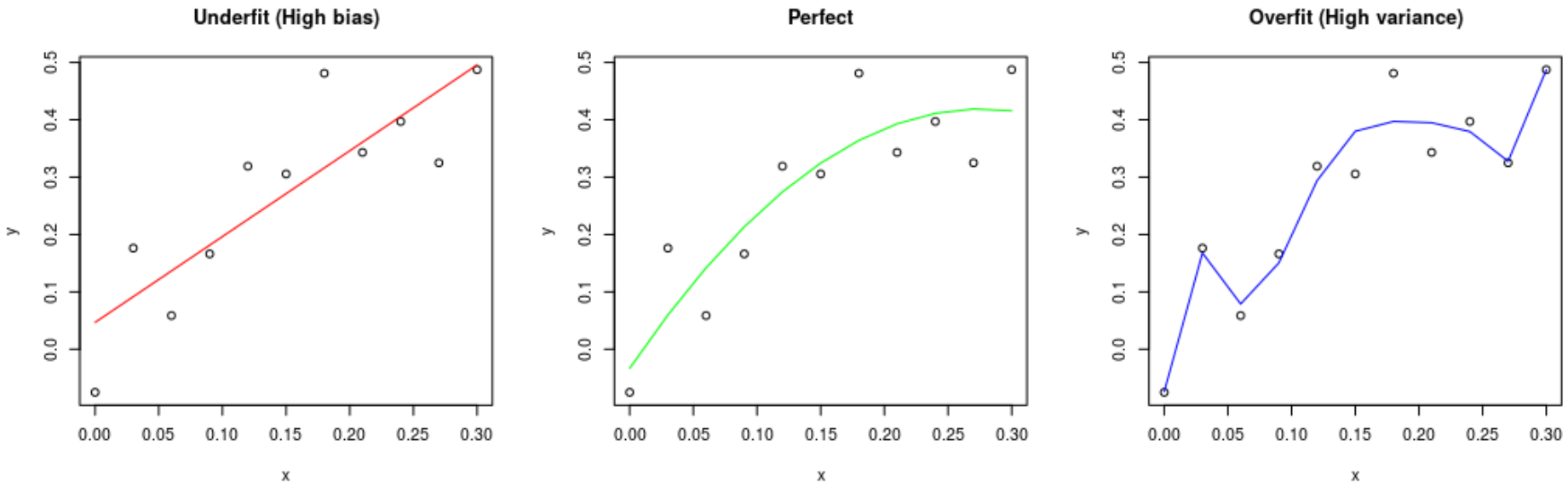

Figure 3: Example of overfitting in regression

Overfitting: Classification

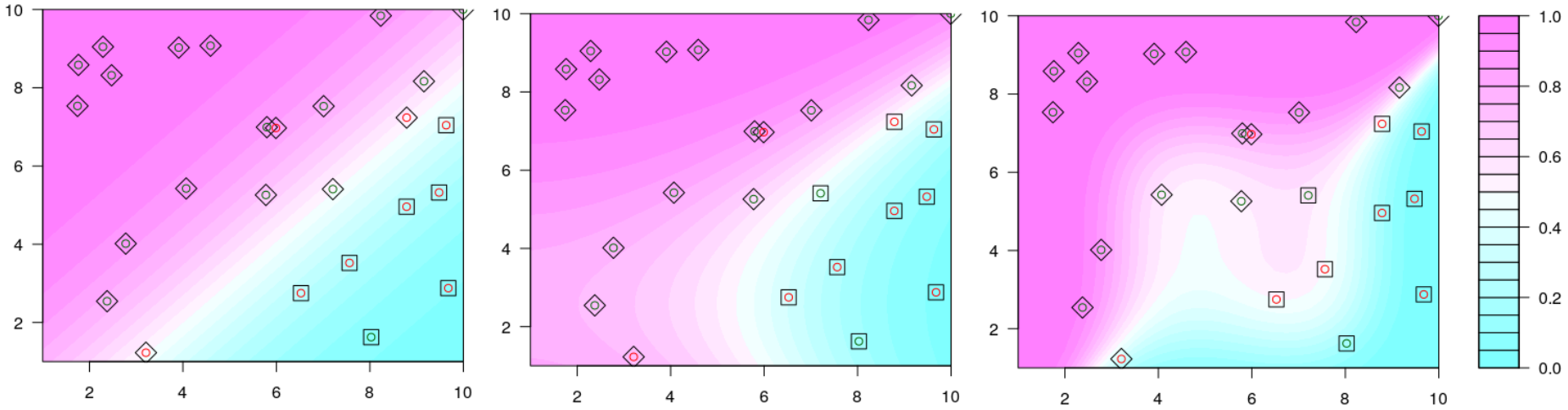

Figure 4: Example of overfitting in classification

Overfitting: Solution?

- **Regularization**:
  - similar to ridge/lasso regression
  - add to cost function *regularisation parameter* $\cdots + \lambda \sum_{j=1} w_j$
  - how to select λ?
- **Early stop**: - stooping learning algorithm before optimal value
  - when?

Training set, test set and test set

- Training set
  - is set used to **train** neural network
- Test set 1
  - is used to select **meta** parameter (number of hidden layers, number of neurons in each layer, λ, ...)
- Test set 2
  - is used to asses overall performance of network

We don't want the model to fit data we have, but underlining relationships in population.

Neural networks in R

- nnets
- neuralnet
  - https://github.com/bips-hb/neuralnet
- keras - tensorflow
  - https://keras.rstudio.com/

Recommended resources

- Deep Learning (Goodfellow)
  - (https://www.deeplearningbook.org/)
- Deep Learning with R (J.J. Allaire)
  - (https://github.com/jjallaire/deep-learning-with-r-notebooks)
- 3Brown1Blue - Series on Neural Networks
  - (https://www.youtube.com/watch?v=aircAruvnKk)
- Coursera: Machine learning (11 week course, matlab)
  - (https://www.coursera.org/learn/machine-learning)
- Neural Network Playground
  - (https://playground.tensorflow.org)