MONASH University
Information Technology

FIT5201
Data Analysis Algorithms

Neural Networks

Outline

- Regularization
- Unsupervised and Self-taught Learning

The power of neural networks

- The model class corresponding to neural networks can represent almost any function (given some minor conditions) provided the network has a sufficiently large number of hidden units
  - Have been widely studied
  - 9 layer can solve many low-level intelligence task pretty well

The power of neural networks

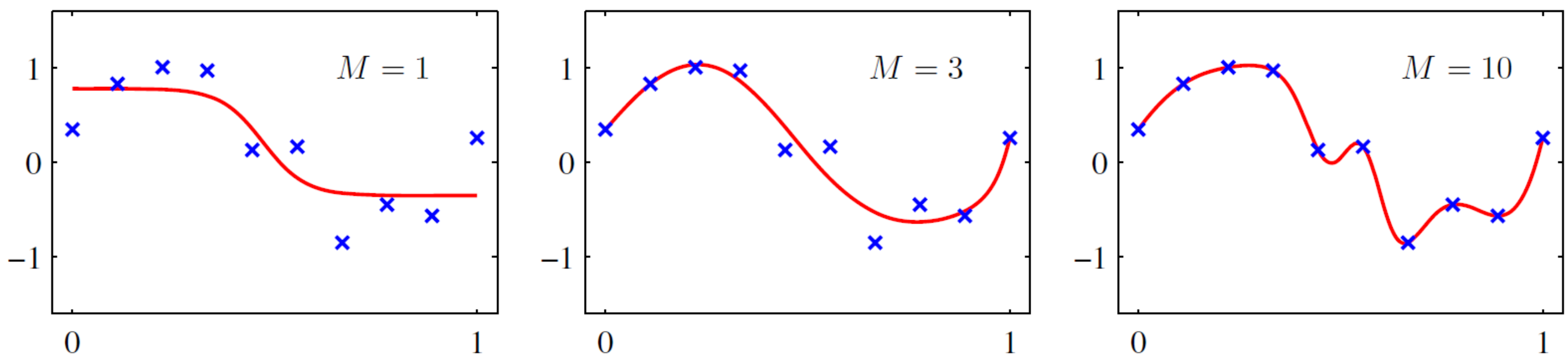
- Classification problem
  - Approximate the target decision boundary to any required precision
- Regression problem:
  - Approximate the target function to any precision
- Price:
  - Large number of neurons in the hidden layers
  - Large number of parameters
  - Tend to overfit the training data

The power of neural networks

- Methods to prevent overfitting
  - Use a large training data
  - Use regularization methods (i.e., weight decay)
  - Use deep architecture instead of wide and shallow architecture
    - Given same number of neurons, deep design performs better
    - Given same performance, deep architecture needs smaller number of neurons
  - Early stopping

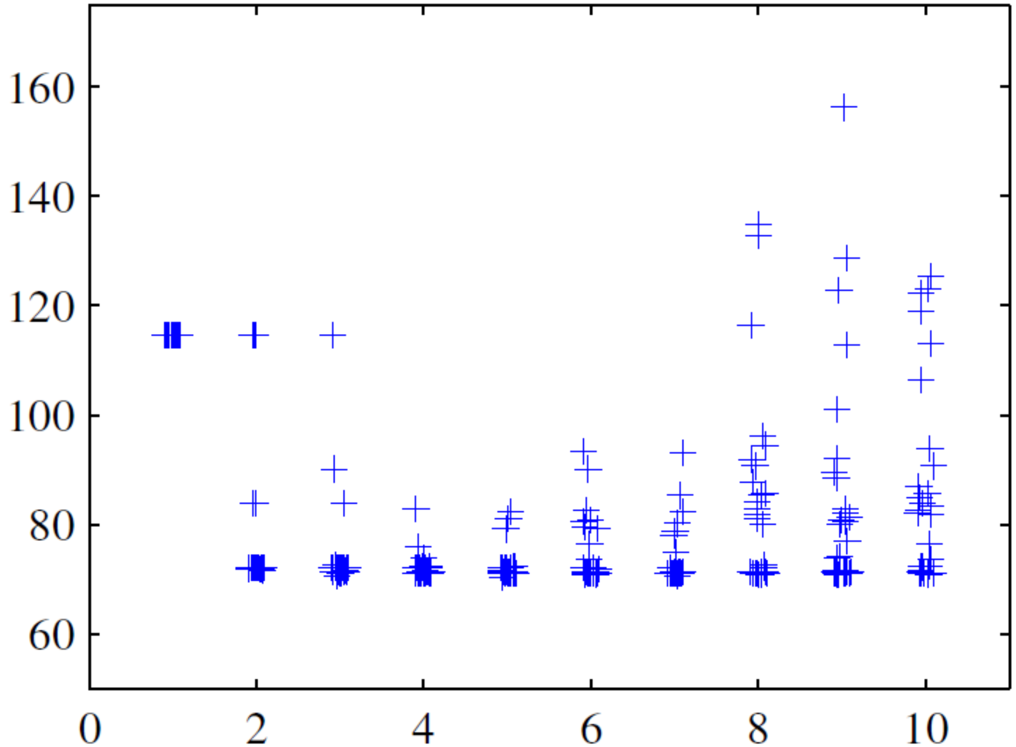
Regularization

- Similar to regression, NNs can overfit the data if too many hidden neurons are added.
  - Blue points: 10 data points drawn from the sinusoidal function plus some noise
  - Red line: the model trained on the neural network with 1 hidden layer
    - > The number of hidden units is different: 1,3, and 10



Regularization…

- The generalization error, however, is not a simple function of M due to the presence of local minima in the error function



1. For each value of M, try 30 random initializations, how?
2. Polynomial data set, sum-of-square validation error
3. Which one is the best M value? How to judge?
4. One way to choose M in practice

Methods to regularize

- Use different initial weights and find the one that has the smallest validation set error
- Weight decay
- Early stopping

Weight Decay

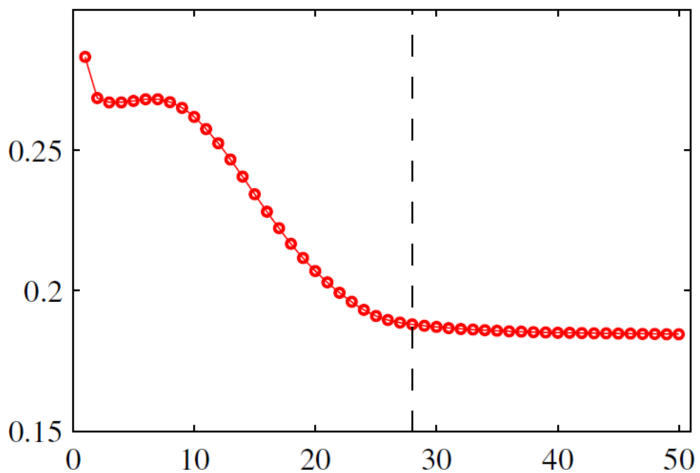
- Similar to polynomial fitting in regression
- Use large number for M and control the complexity by using a regularization term to the error function
- The effective model complexity is then determined by the choice of the regularization coefficient λ

Early Stopping

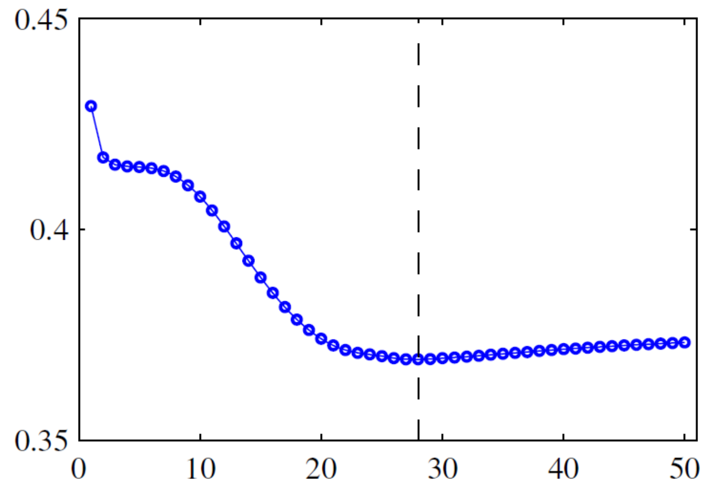
- Stop the training when the network is not learning anymore

Early Stopping

- Stop the training when the network is not learning anymore
- Stop at the point where the validation set error starts to increase
  - Indicates overfitting



Training error

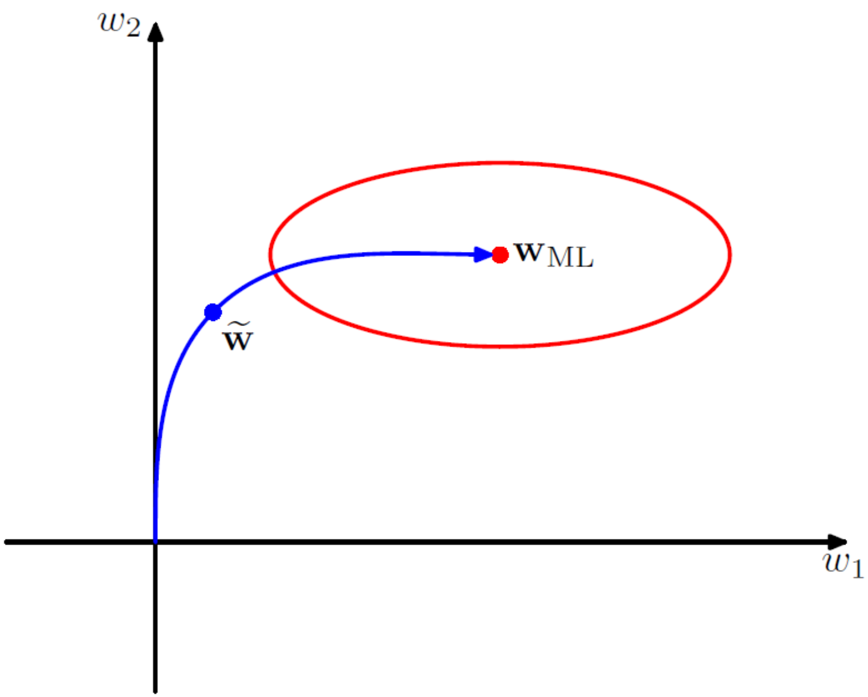


Test error

Prechelt, Lutz; Genevieve B. Orr (2012-01-01). “Early Stopping - But When?”. In Gregoire Montavon, Klaus-Robert Muller (eds.) Neural Networks: Tricks of the Trade. Lecture Notes in Computer Science. Springer Berlin Heidellberg. Pp. 53-67. ISBN 978-3-642-35289-8. Retrieved 2013-12-15

Early Stopping …

- Has the same effect as weight decay
  - Has theoretical support
  - Intuitive explanations

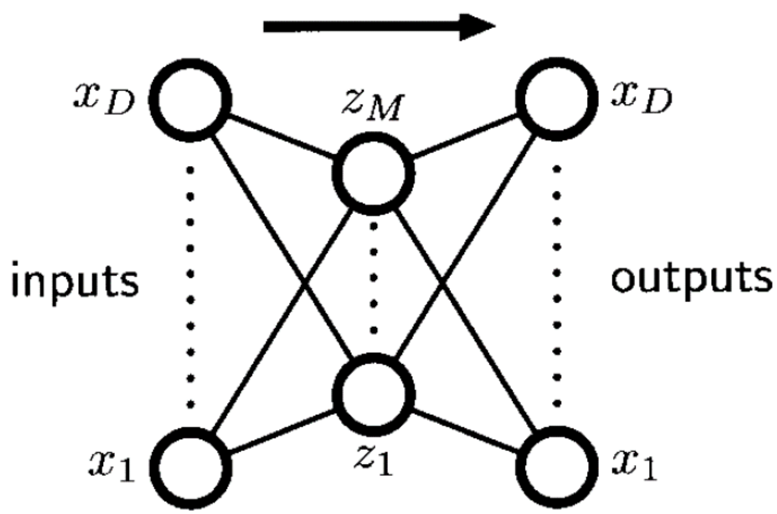


Unsupervised Learning with NN

- Neural networks are generally considered supervised learning
- In unsupervised learning no model output is present
- Important component in deep learning
- Unsupervised learning:
  - Clustering
  - Dimension reduction
  - Encoding
  - …..

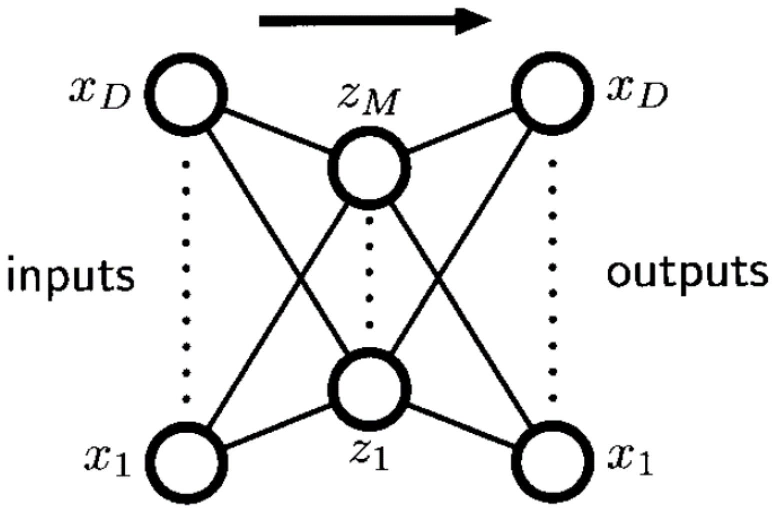
Unsupervised Learning with NN

- Connect the output back to the input and create a loop
  - Number of output neurons is the same as the number of input neurons
  - Optimize the weights so that the input pattern is also at the output
  - Creates an unsupervised neural network



Autoencoder

- Unsupervised NN can be considered as dimensionality reduction



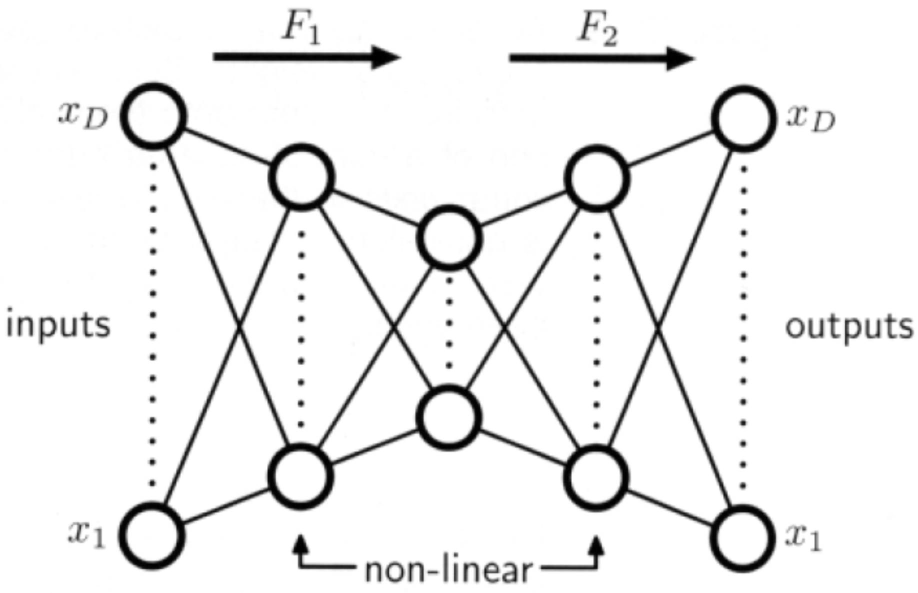
Minimize $E(\boldsymbol{\theta}) := \frac{1}{2}\sum_{n=1}^{N} ||h_{\boldsymbol{\theta}}(\boldsymbol{x}_n) - \boldsymbol{x}_n||_2^2$

PCA vs Autoencoder

- If the transformations in the NN is linear, Autoencoder is identical to PCA.
  - They both minimize linear models using the same sum of squares method
  - For linear models, PCA is more efficient than Autoencoder
- Auto encoder can do non-linear transformations but PCA cannot.
  - One hidden layer solution can be given by the projection onto the principal component subspace
  - Need extra hidden layer

Autoencoder …

- F_1 is the encoder and F_2 is the decoder



Visualization of Autoencoder

- Visualize what one hidden unit encodes
- How?
- What's visualizable?
  - The input
- For each input, one hidden unit will generate one value
- Try all the inputs to see which one can maximize this hidden unit.
  - Expensive to do this
- Solutions:
  - Without running on all the inputs, can we just compute with equations?

Visualization of Autoencoder

$$a_i^{(2)} = f\left(\sum_{j=1}^{100} W_{ij}^{(1)} x_j + b_i^{(1)}\right)$$

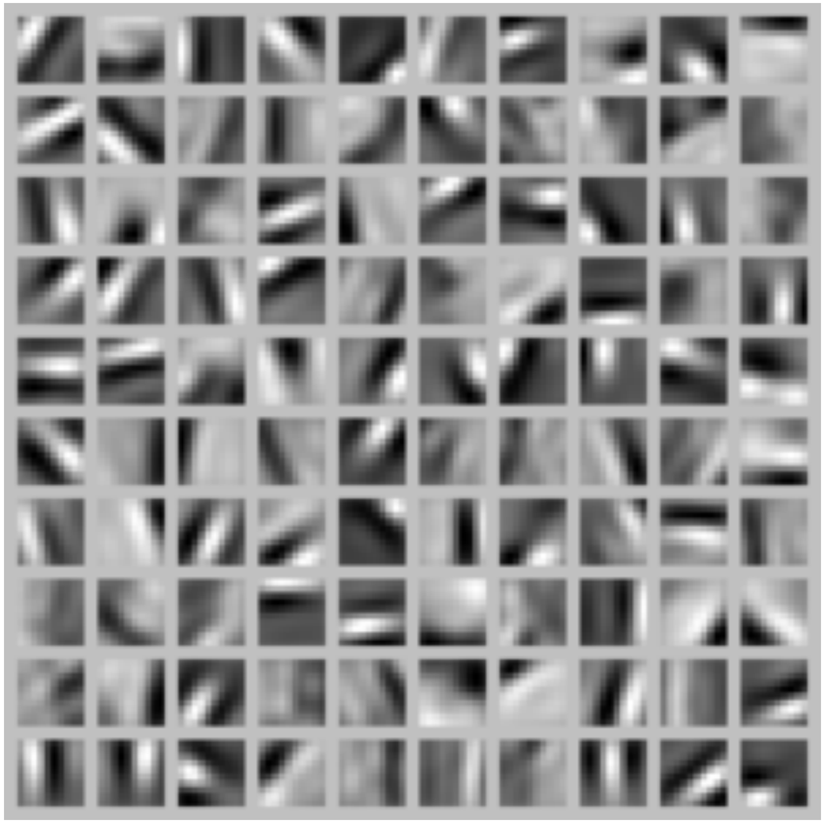
1. Treat x_j as variable, W and b as constants
2. Maximize this function

$$x_j = \frac{W_{ij}^{(1)}}{\sqrt{\sum_{j=1}^{100} (W_{ij}^{(1)})^2}}$$

with the constraint: $||\boldsymbol{x}|| \leq 1$

Visualization of Autoencoder

- An autoencoder on images
- Visualize the $\boldsymbol{x}$ that maximizes each hidden unit.

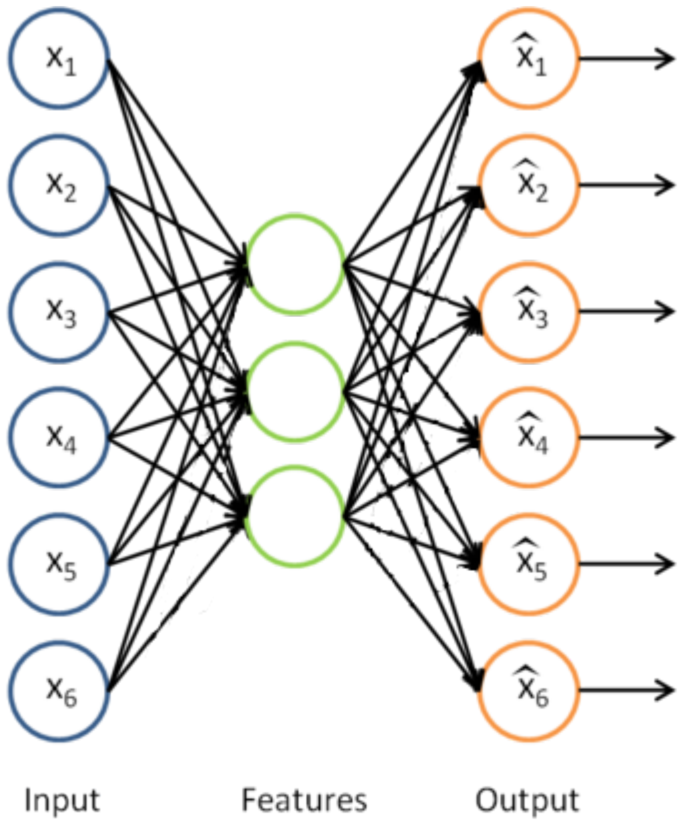


Self-Taught Learning with NNs

- If you have limited labels
- Semi-supervised learning
- How to make use of unlabelled data

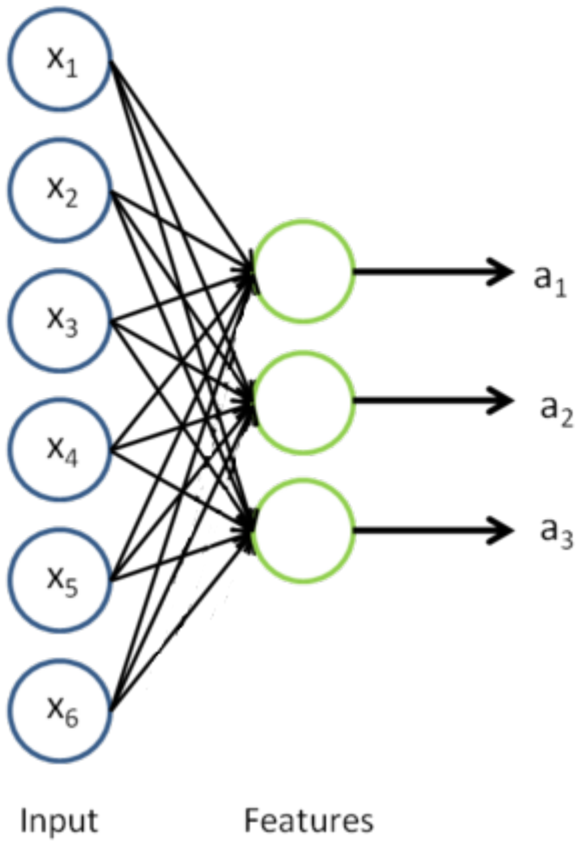
Self-Taught Learning with NNs

- Learn the features from the autoencoder



Self-Taught Learning with NNs

- Trained encoder can now be used in training a classifier



1. Use $\boldsymbol{a}$ to replace the original input $\boldsymbol{x}$
2. Concatenate $\boldsymbol{a}$ and $\boldsymbol{x}$, and use the concatenation to replace $\boldsymbol{x}$

Want to know more about NNs