Neural Network

Fujian Yan

Assistant Teaching Educator
School of Computing, Wichita State University
Office: Jabara Hall 222
Office (student) hours: M 13:00-15:00
fujian.yan@wichita.edu

Outlines

- Midterm Presentation
- Neural Network

PRESENTATIONS

Tentative Presentation Date

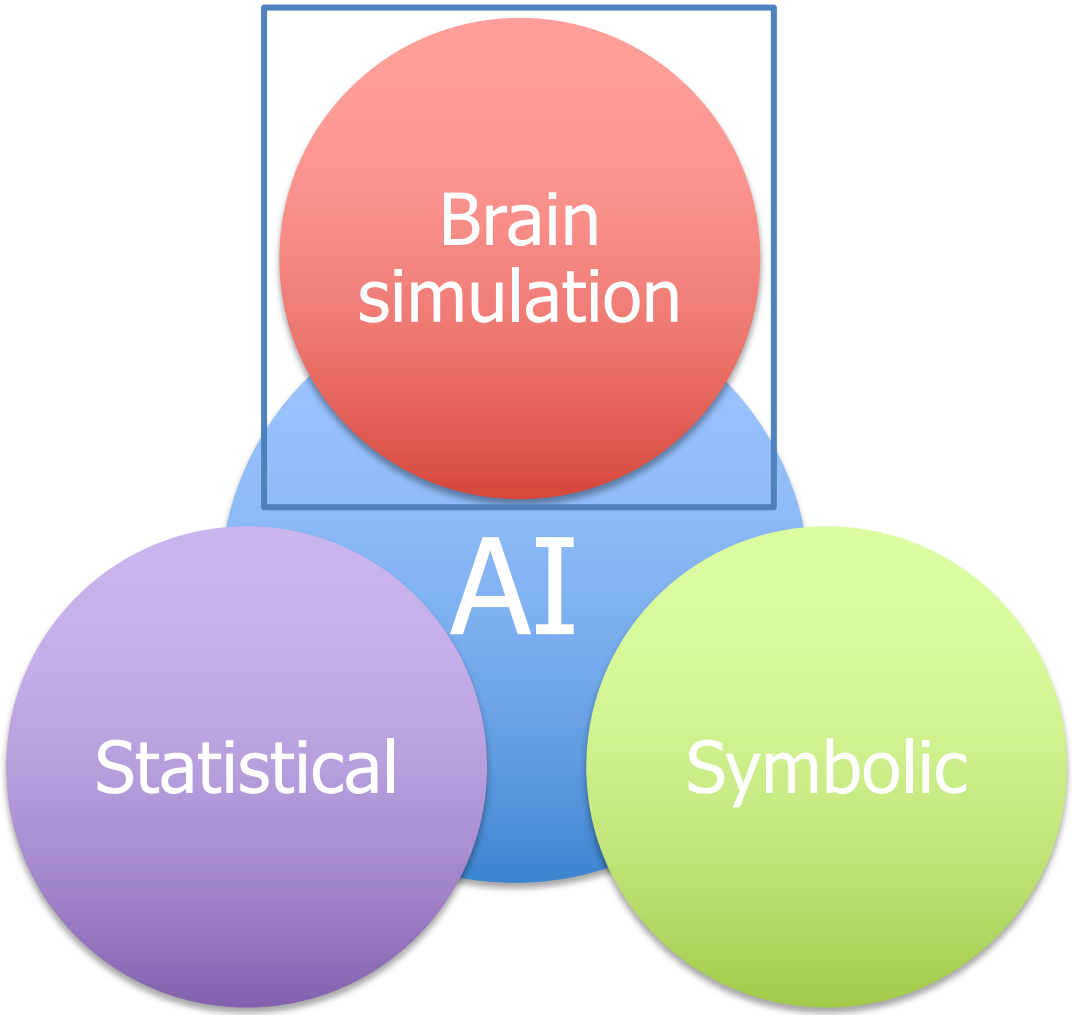
Presentation Name	Date
Project Proposal Presentation	02/14/2023; 02/16/2023; 02/21/2023
Midterm Presentation	03/07/2023; 03/09/2023; 03/21/2023
Check Up presentation 1	03/30/2023
Check Up presentation 2	04/13/2023

Midterm Presentation

- Midterm Presentation
  - 20 min = 15 min + 5 min QA
  - **All students** should show up during the presentation
  - Preliminary results
    - Clearly describe the dataset
    - Progress of current project
    - Related Work (each student should present on paper regarding the topic)
    - Contribution of each group member based on the task distribution
    - Problems if you have any
  - Need to turn in the slides and the data **Dashboard**

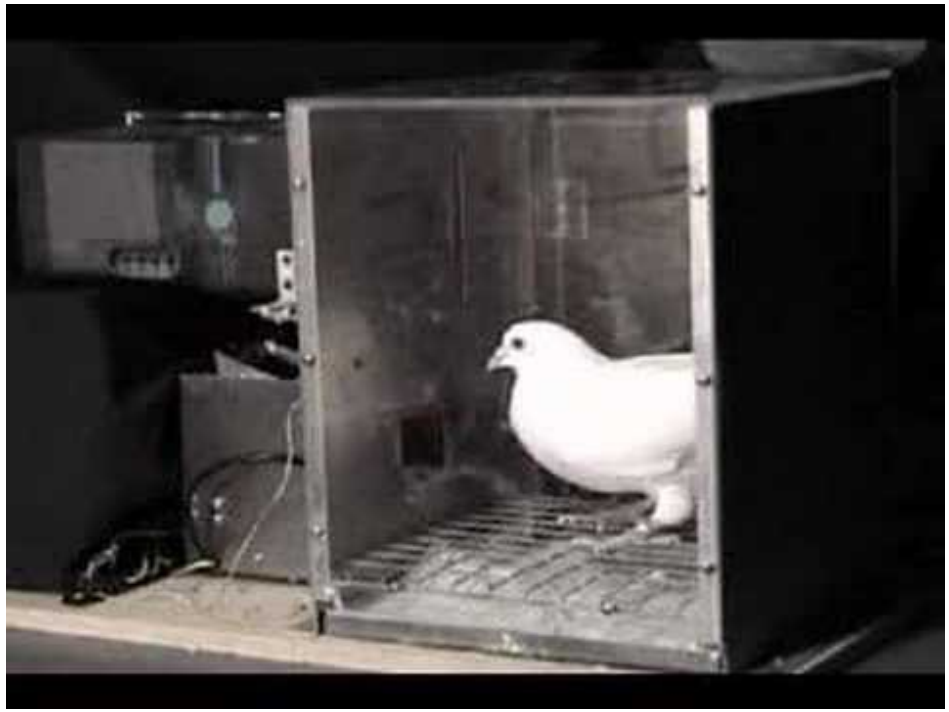
Questions?

AI Approaches



Experiment: Pigeon in Skinner box

- Pigeons as art experts (Watanabe. et al. 1995)
  - Present paintings of two different artists (e.g., Chagall/Van Gogh)
  - Reward the pigeon for pecking when it is presented a particular artist (e.g., Van Gogh)



Experiment: Pigeon in Skinner box

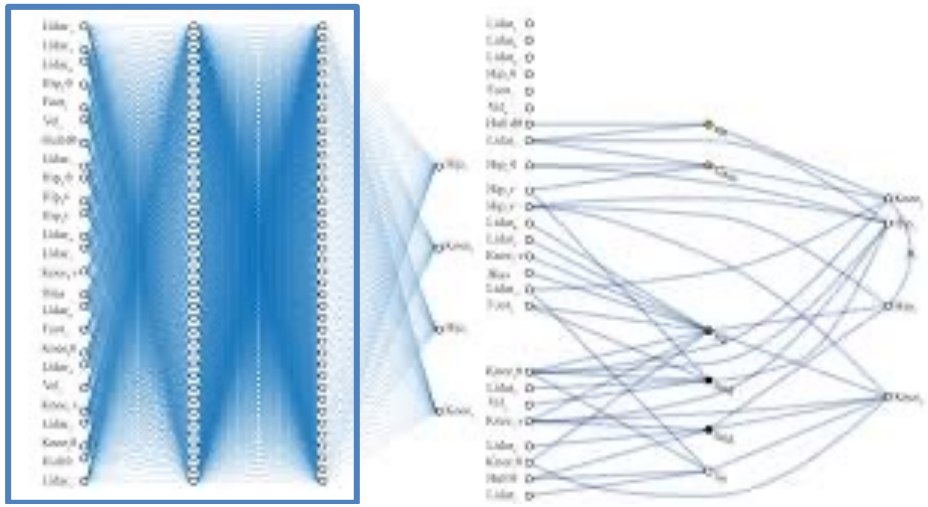
- When presented with pictures they had been trained on, pigeons could tell the difference between Van Gogh and Chagall accuracy

  95%
- When presented previously unseen paintings of the artists

  85%

What is a Neural Network (NN)

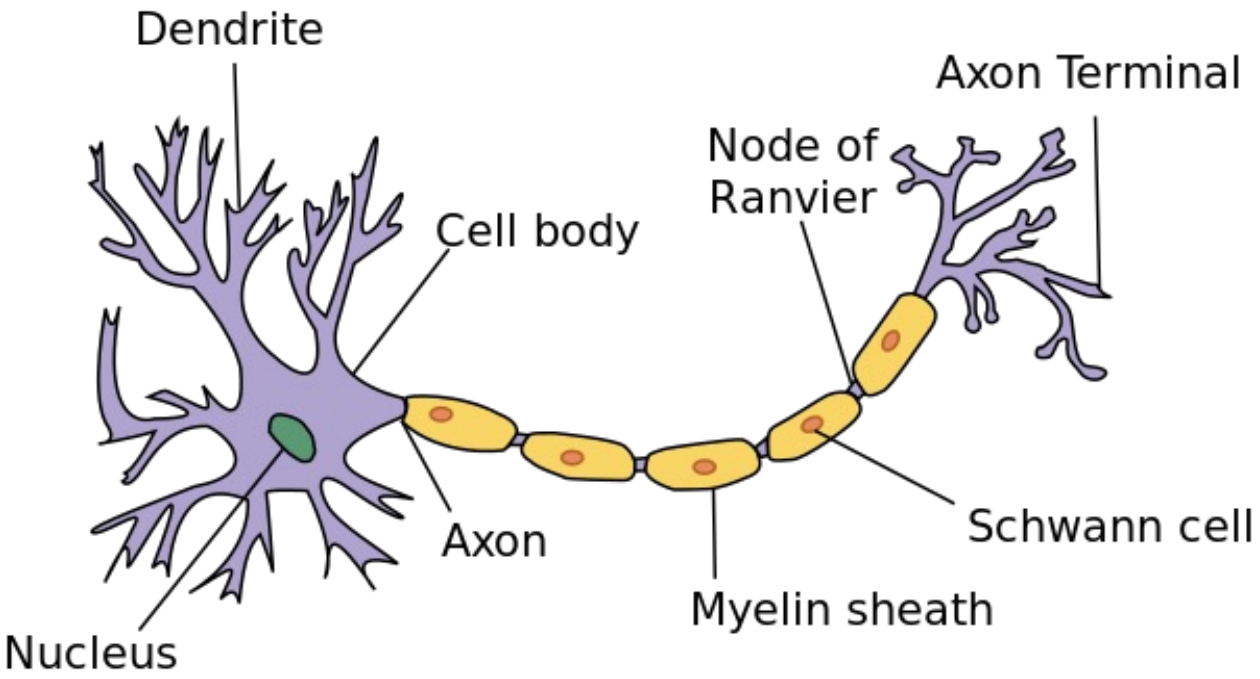
- A neural network is a *massively parallel distributed processor* made up of simple processing unit, which has a natural propensity for *storing experiential knowledge* and making it available for use.
- The artificial neural networks are largely inspired by the biological neural networks, and the ultimate goal of building an intelligent machine which can mimic the human brain.



Biological Neurons

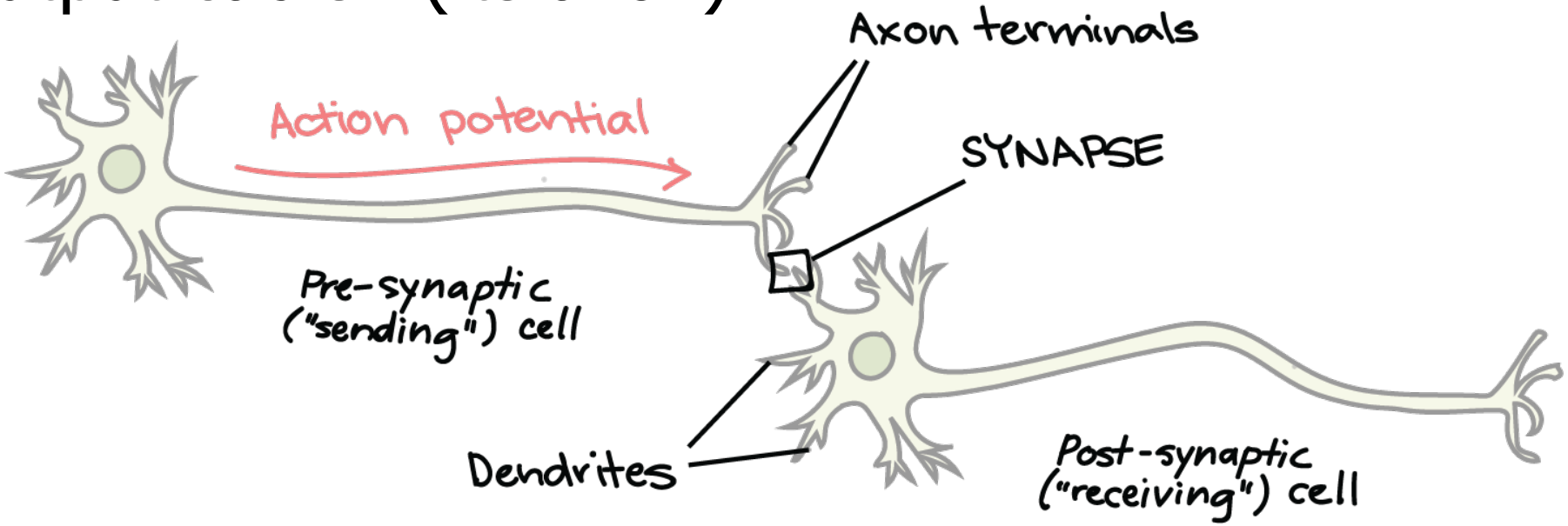
A typical biological neuron is composed of

- A cell body
- Hair-like dendrites: input channels
- Axon: output cable; it usually branches.



Biological Neurons

- The neuron responds to many sources of electric impulses in three ways: some inputs excite the neuron, some inhibit it, and some modulate its behavior.
- If the neuron becomes sufficiently excited, it responds ("fires") by sending an electric pulse (a spike)-down its output cable—(its axon).



Biological Neuron

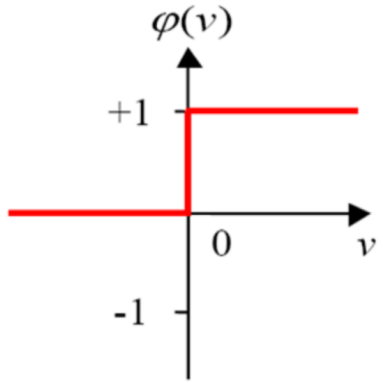
- It receives information, usually in the form of electrical pulses, from many other neurons.
- It does what is, in effect, a complex dynamic sum of these inputs.
- It sends out information in the form of a stream of electrical impulses down its axon and on to many other neurons.
- The conduction of nerve impulses is all-or-none.
- The connections (synapses) are crucial for excitation, inhibition or modulation of the cells.
- Learning is possible by adjusting the synapses.

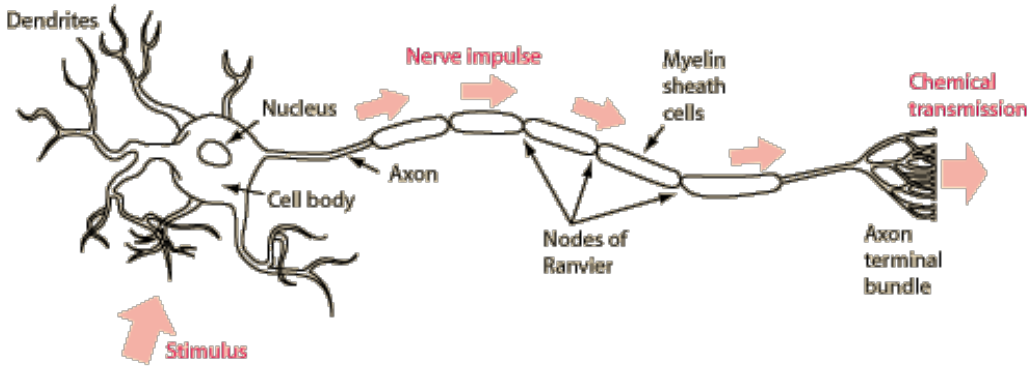
NEURAL NETWORK MODEL

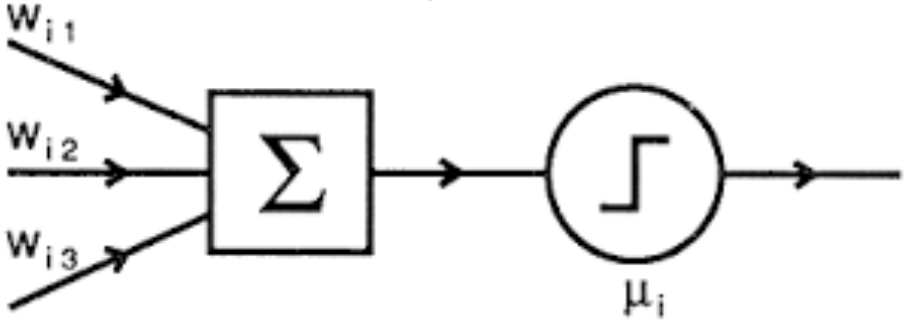
NN Model

- Threshold: The neuron will not fire till it is “high”
- Nonlinear activation function
- Inhibition and excitation: synaptic weights

$$y = \varphi\left(\sum_{i=1}^{m} x_i - b\right)$$







Benefits of Neural Network

- High computational power
  - *Generalization*: Producing reasonable outputs for inputs not encountered during training (learning).
  - Has a massively parallel distributed structure.
- Useful properties and capabilities
  - Nonlinearity: Most physical systems are nonlinear.
  - Adaptivity (plasticity): Has built-in capability to adapt their synaptic weights to changes in the environment.
  - Fault tolerance: If a neuron or its connecting links are damaged, the overall response may still be ok (due to the distributed nature of information stored in a network).

LEARNING WEIGHTS

Learning Weights

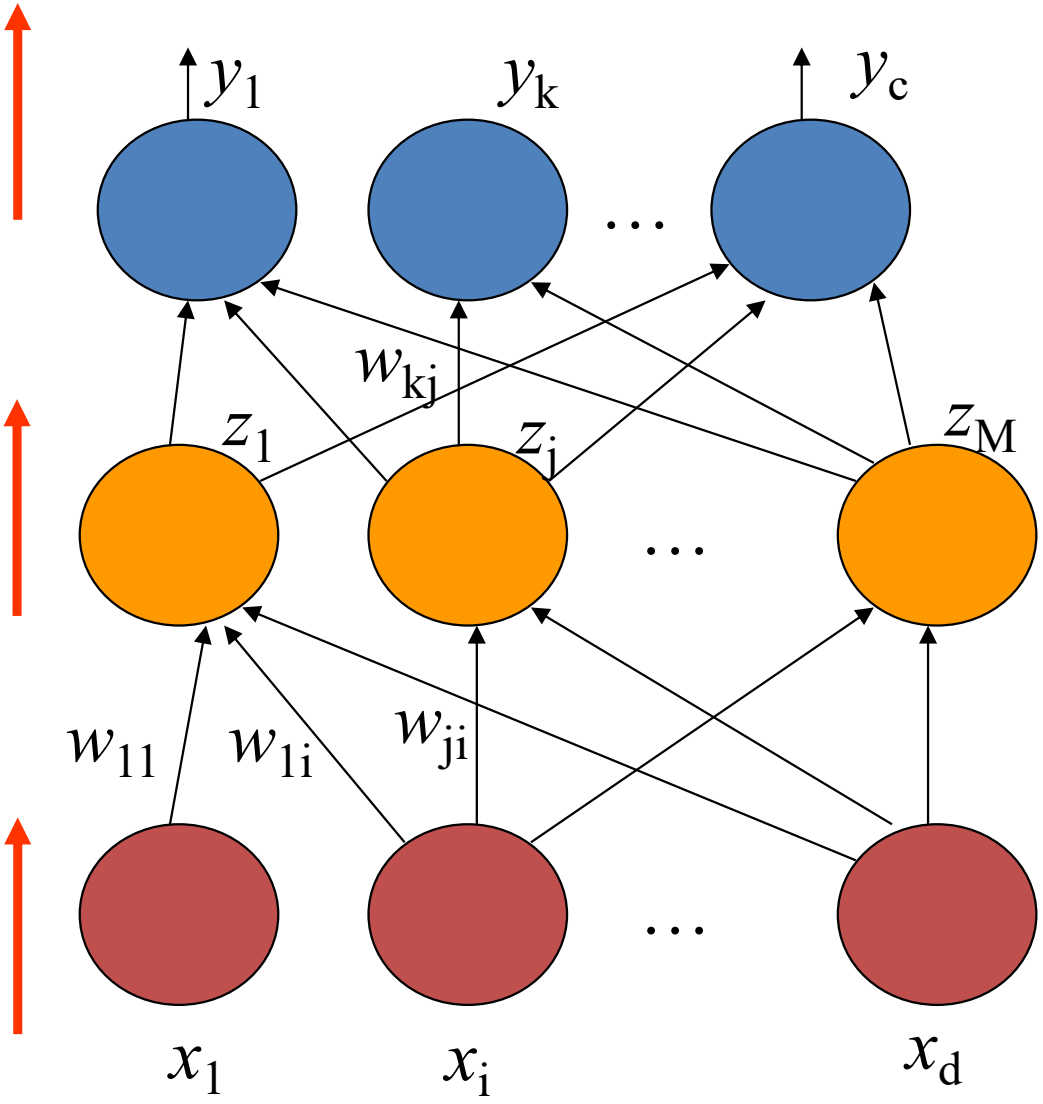
- How to adjust weights?
- Adjust weights using known examples (training data) $(x_1,x_2,x_3,\ldots,x_d,t)$.
- Try to adjust weights so that the difference between the output of the neural network y and t (target) becomes smaller and smaller.
- Goal is to minimize Error (difference)

Neural Network Learning: Two Processes

- Forward propagation: present an example (data) into neural network. Compute activation into units and output from units.
- Backward propagation: propagate error back from output layer to the input layer and compute derivatives (or gradients).

Forward Propagation

- L2: Activation function: *f* (linear, sigmoid, softmax)
- Activation $\sum_{j=0}^{M} w_{kj} z_j$
- L1: Activation function: *g* (linear, tanh, sigmoid)
- Activation $\sum_{i=0}^{d} w_{ji} x_i$



Backward Propagation

$$E = \frac{1}{2}\sum_{k=1}^{C}(y_k - t_k)^2$$

$$\frac{\partial E}{\partial y_k} = y_k - t_k$$

$$\frac{\partial E}{\partial a_k} = \frac{\partial E}{\partial y_k}\frac{\partial y_k}{\partial a_k} = (y_k - t_k) f'(a_k) = \delta_k$$

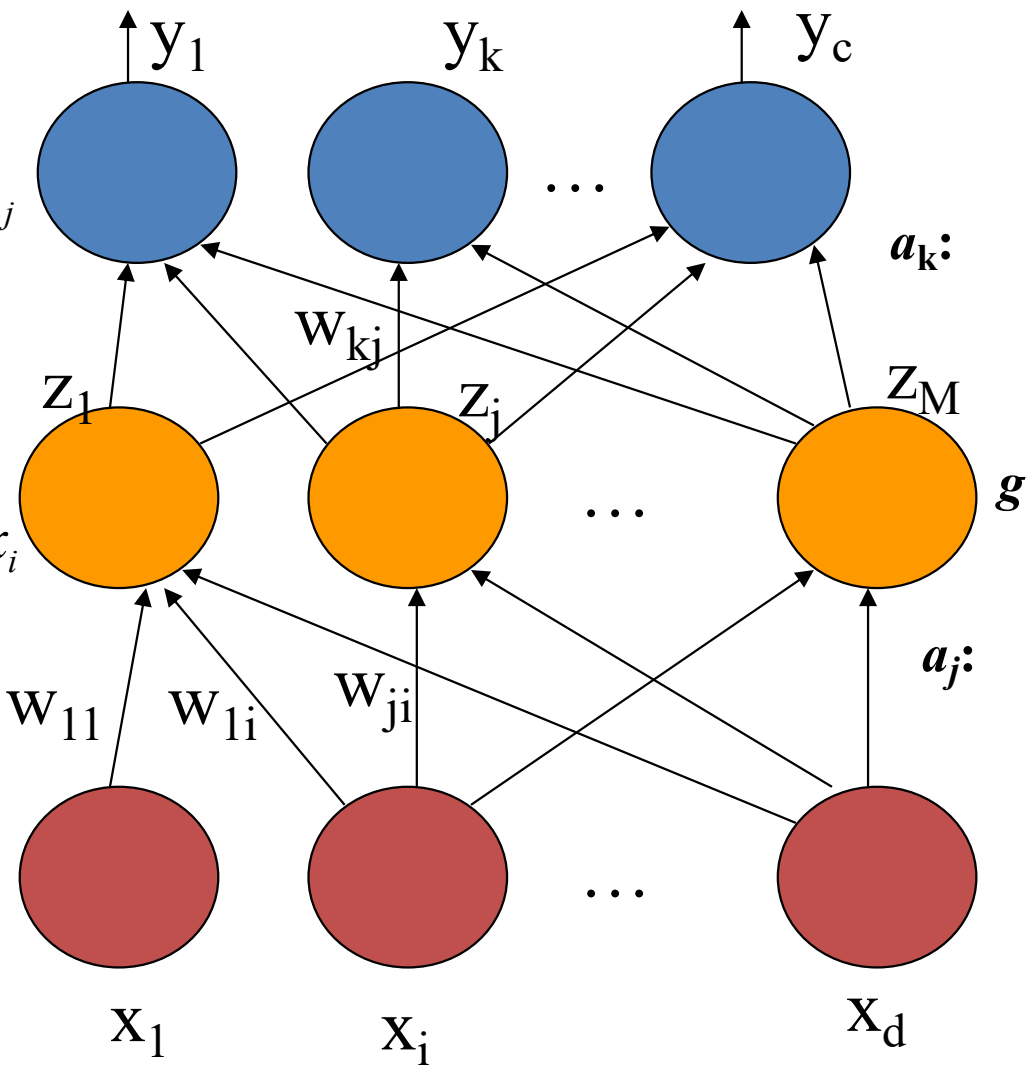
$$\frac{\partial E}{\partial w_{kj}} = \frac{\partial E}{\partial a_k}\frac{\partial a_k}{\partial w_{kj}} = \delta_k z_j$$

$$\frac{\partial E}{\partial a_j} = \sum_{k=1}^{c}\frac{\partial E}{\partial y_k}\frac{\partial y_k}{\partial a_k}\frac{\partial a_k}{\partial z_j}\frac{\partial z_j}{\partial a_j} = \sum_{k=1}^{C}\delta_k w_{kj} g'(a_j) = \delta_j$$

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial a_j}\frac{\partial a_j}{\partial wji} = \delta_j x_i$$

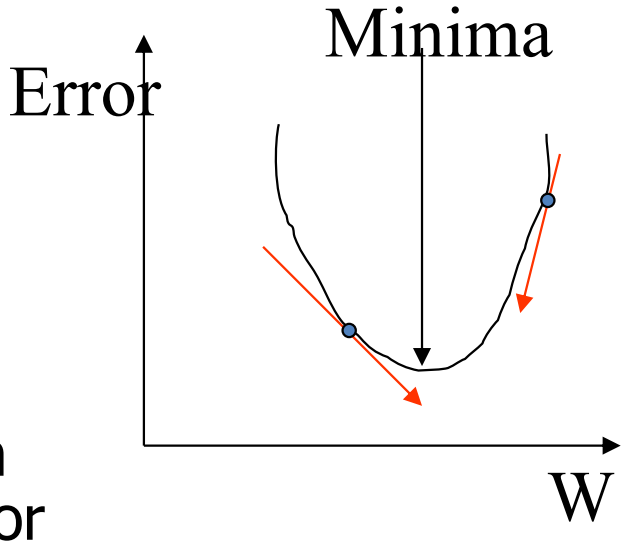
$\boldsymbol{a_k}$: $\sum_{j=1}^{M} w_{kj} z_j$

$\boldsymbol{a_j}$: $\sum_{i=1}^{d} w_{ji} x_i$



Learning by Gradient Decent (Back-Propagation)

- **Initialize** weights *w*
- **Repeat**
  - For each data point *x*, do the following:
  - Forward propagation: compute outputs and activations
  - Backward propagation: compute errors for each output units and hidden units. Compute gradient for each weight.
  - Update weight $w \mathrel{*}= w - \eta\,(\partial E / \partial w)$
- **Until** a number of iterations or errors drops below a threshold.



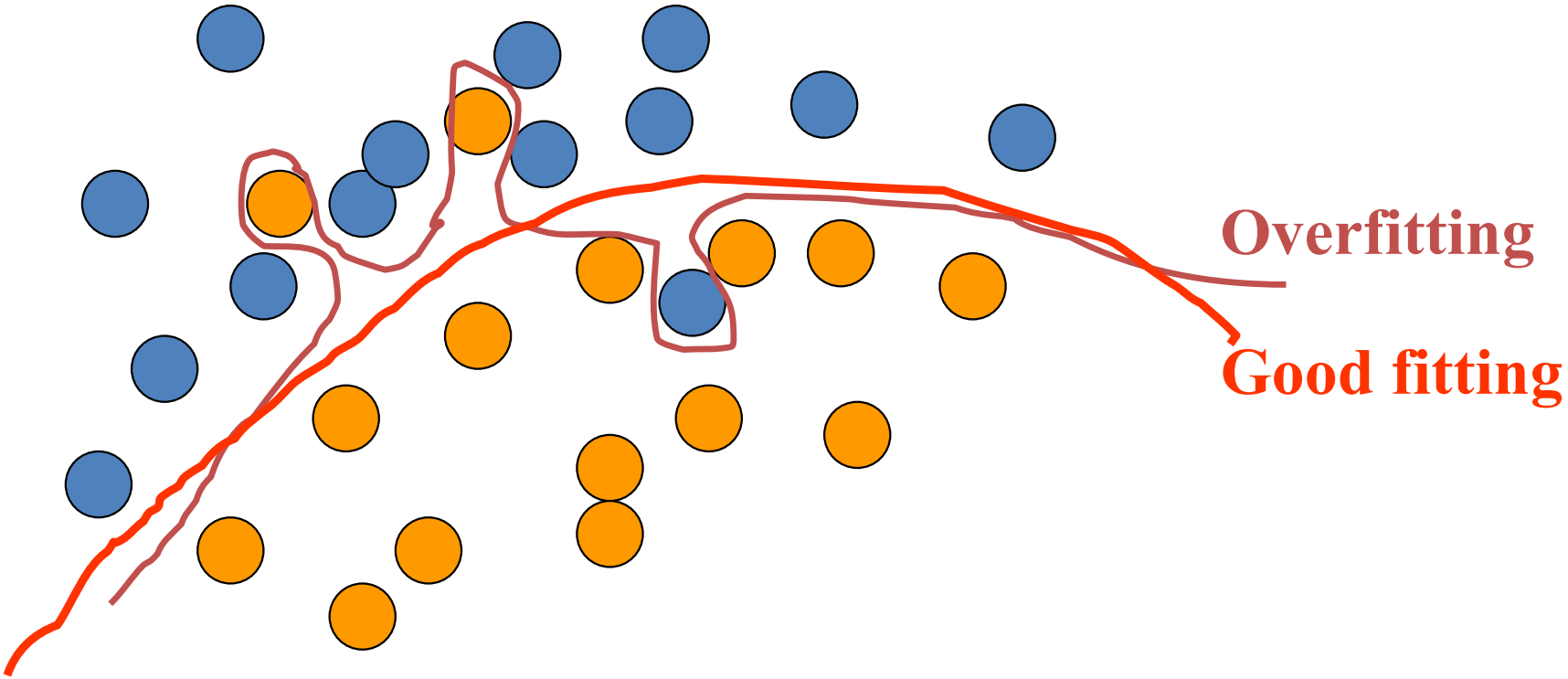
Note: η is learning rate or step size.

Overfitting in Learning

- The training data contains information about the regularities in the mapping from input to output. But it also contains noise
  - The target values may be unreliable.
  - There is sampling error. There will be accidental regularities just because of the particular training cases that were chosen.
- When we fit the model, it cannot tell which regularities are real and which are caused by sampling error.
  - So it fits both kinds of regularity.
  - If the model is very flexible it can model the sampling error really well. This is a disaster.

Example of Overfitting and Good Fitting

- Overfitting function can not generalize well to unseen data.

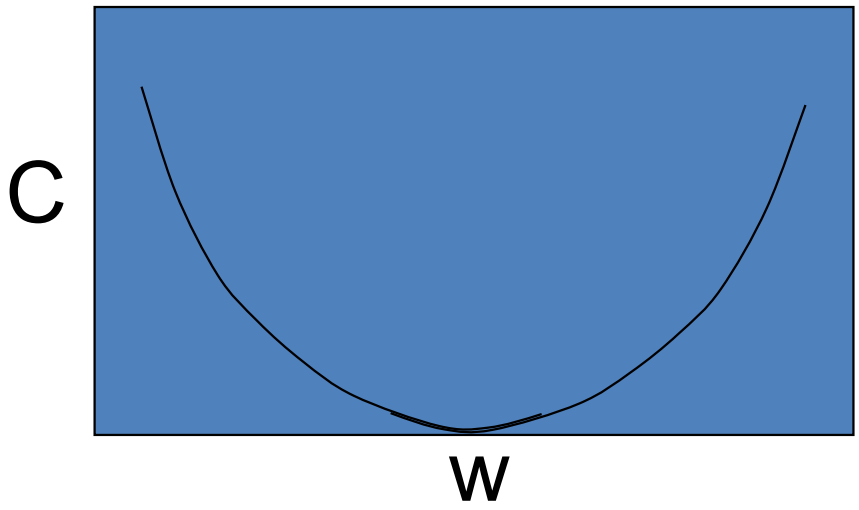


Preventing Overfitting

- Use a model that has the right capacity:
  - enough to model the true regularities
  - not enough to also model the spurious regularities (assuming they are weaker).
- Standard ways to limit the capacity of a neural net:
  - Limit the number of hidden units.
  - Limit the size of the weights.
  - Stop the learning before it has time to overfit.

Limiting the Size of the Weights

- Weight-decay involves adding an extra term to the cost function that penalizes the squared weights.
  - Keeps weights small unless they have big error derivatives.



$$C = E + \frac{\lambda}{2}\sum_i w_i^2$$

$$\frac{\partial C}{\partial w_i} = \frac{\partial E}{\partial w_i} + \lambda w_i$$

$$when \quad \frac{\partial C}{\partial w_i} = 0, \quad w_i = -\frac{1}{\lambda}\frac{\partial E}{\partial w_i}$$

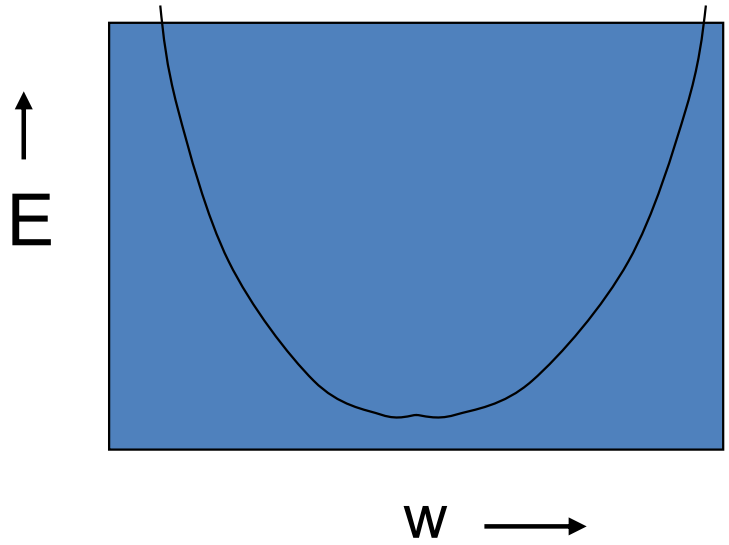
Using a Validation Set

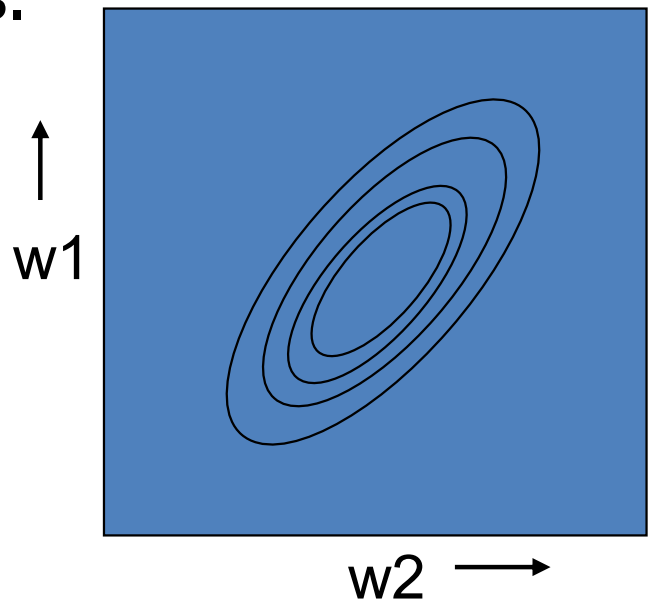
- Divide the total dataset into three subsets:
  - Training data is used for learning the parameters of the model.
  - Validation data is not used of learning but is used for deciding what type of model and what amount of regularization works best.
  - Test data is used to get a final, unbiased estimate of how well the network works. We expect this estimate to be worse than on the validation data.
- We could then re-divide the total dataset to get another unbiased estimate of the true error rate.

TRAINING ISSUES

How to Speedup Learning?

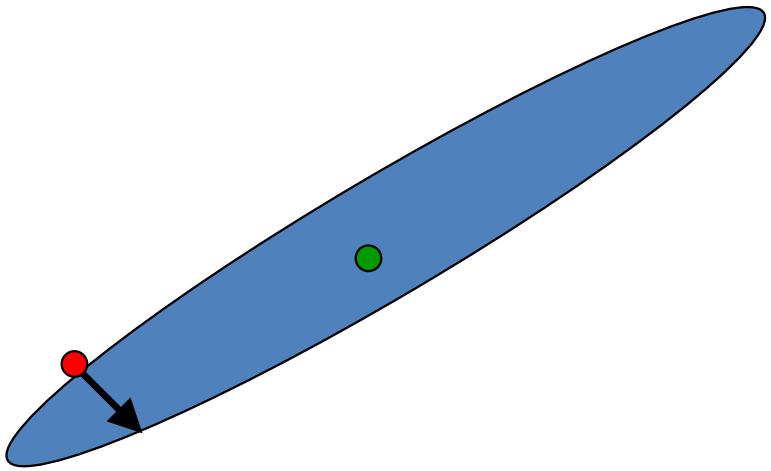
- The error surface lies in a space with a horizontal axis for each weight and one vertical axis for the error.
  - It is a quadratic bowl: the height can be expressed as a function of the weights without using powers higher than 2. Quadratics have constant curvature (because the second derivative must be a constant)
  - Vertical cross-sections are parabolas.
  - Horizontal cross-sections are ellipses.





Convergence Speed

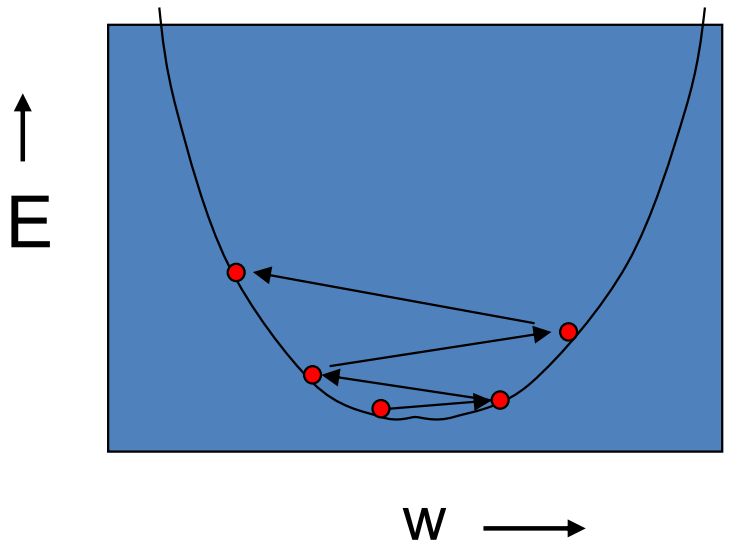
- The direction of steepest descent does not point at the minimum unless the ellipse is a circle.
  - The gradient is big in the direction in which we only want to travel a small distance.
  - The gradient is small in the direction in which we want to travel a large distance.



$$\Delta w_i = -\varepsilon \frac{\partial E}{\partial w_i}$$

How the Learning Goes Wrong

- If the learning rate is big, it sloshes to and from across the ravine. If the rate is too big, this oscillation diverges.
- How can we move quickly in directions with small gradients without getting divergent oscillations in directions with big gradients?



Five Ways to Speed up Learning

- Use an adaptive global learning rate
  - Increase the rate slowly if it's not diverging
  - Decrease the rate quickly if it starts diverging
- Use separate adaptive learning rate on each connection
- Use momentum
  - Instead of using the gradient to change the position of the weight "particle", use it to change the velocity.
- Use a stochastic estimate of the gradient from a few cases
  - This works very well on large, redundant datasets.

Challenges with Neural Network

- Model
  - Vanishing gradients
  - Cannot use unlabeled data
  - Hard to understand the relationship between input and output
  - Cannot generate data
- Training
  - Diminishing gradient inhibits multiple layers
  - Can get stuck in local minimums
  - Training time can be extensive

HPC

- https://www.wichita.edu/services/hpc/hpc-guides.php

Questions?

Summary

- Review the plan of this semester
- Dataset Preview
- Form a team next week by Tuesday.
- Project Selection