Training Neural Networks

Deep Learning
What society thinks I do
What my friends think I do
What other computer scientists think I do
What mathematicians think I do
What I think I do
What I actually do

```
from theano import *
```

SML310: Research Projects in Data Science, Fall 2019

Michael Guerzhoy

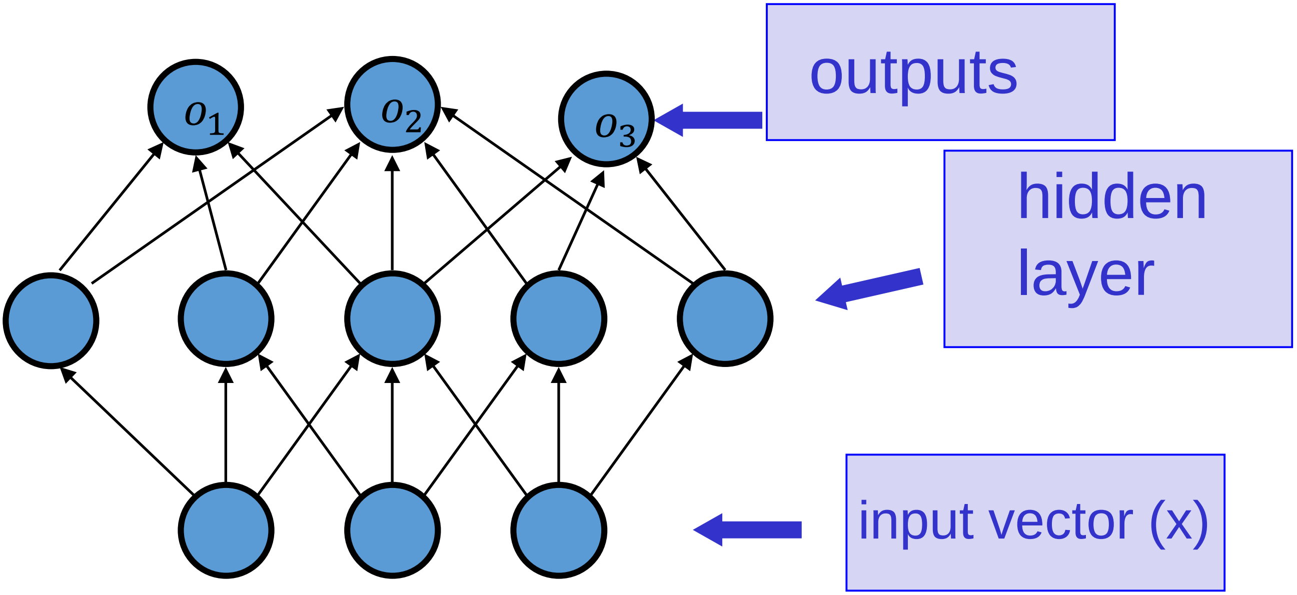
One-Hot Encoding for Inputs and Outputs

$$\rightarrow \begin{bmatrix}0\\1\\0\\0\\0\\0\end{bmatrix}$$

$$\rightarrow \begin{bmatrix}0\\0\\0\\0\\0\\1\end{bmatrix}$$

Multilayer Neural Network for Classification

o_i is large if the probability that the correct class is *i* is high



outputs

hidden layer

input vector (x)

A possible cost function:

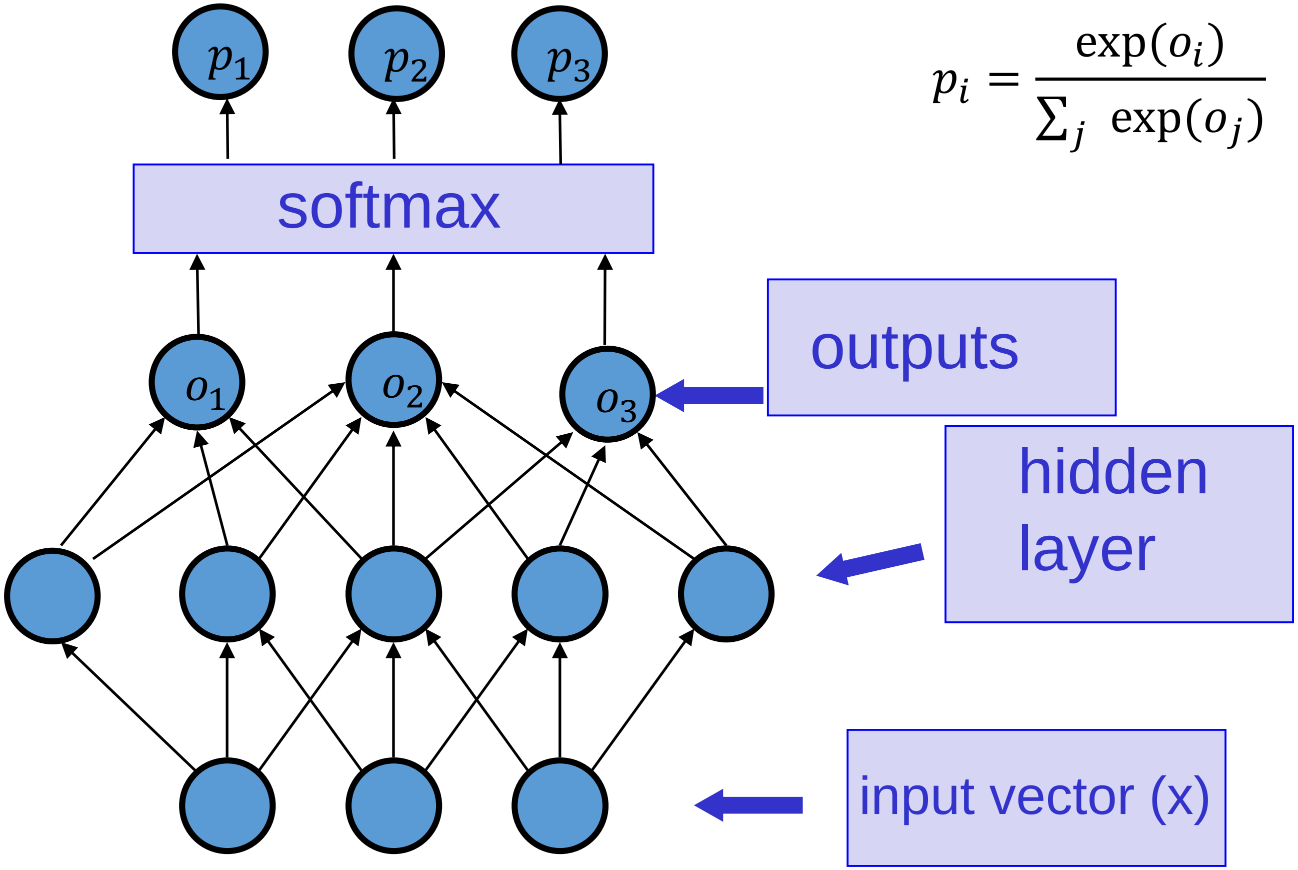
$$\sum_{i=1}^{m}\left(o^{(i)} - y^{(i)}\right)^2$$

$y^{(i)\prime}s$ encoded using one-hot encoding

Softmax

- Want to estimate the probability $P(y = y'|x, \theta)$
  - θ: network parameters

$$p_i = \frac{\exp(o_i)}{\sum_j \exp(o_j)}$$



softmax

outputs

hidden layer

input vector (x)

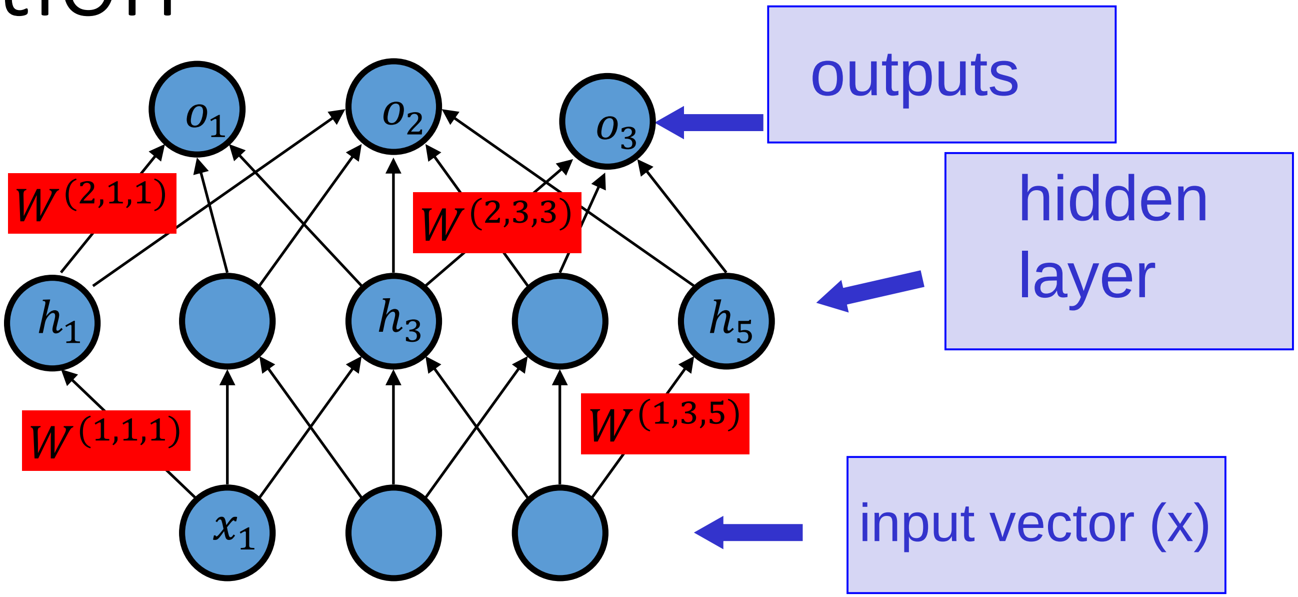
Softmax

- $p_i = \frac{\exp(o_i)}{\sum_j \exp(o_j)}$ can be thought of as probabilities
  - $0 < p_i < 1$
  - $\sum_j p_j = 1$
  - This is a generalization of logistic regression
    - (For two outputs, $p_1 = \frac{\exp(o_1)}{\exp(o_1)+\exp(o_2)} = \frac{1}{1+\exp(o_2-o_1)}$)

Cost Function: $-\sum_j \ y_j log p_j$

- Likelihood (single training case): $P(y_j = 1; x|w)$
  - The probability for $y_j = 1$ that the network outputs with weights w
- The likelihood of $y = (0, \dots, 0, 1, 0, 0, \dots, 0)$ is p_j, where j is the index of the non-zero entry in y
  - Same as $\Pi_j p_j^{y_j}$
- Negative log-likelihood (single training case)
  - $-\sum_j \ y_j log p_j$

Vectorization



- $o_i = g\left(\sum_j W^{(2,j,i)} h_j + b^{(2,j)}\right)$
- So o $= g\left(\left(W^{(2)}\right)^T h + b^{(2)}\right)$
- Similarly, h = $g\left(\left(W^{(1)}\right)^T x + b^{(1)}\right)$