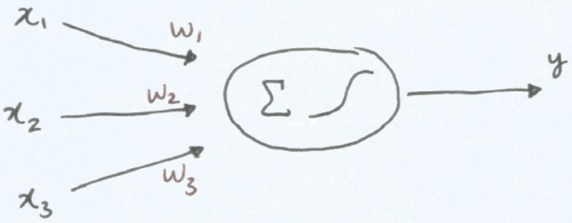
WEEK 9.2

NEURAL NETWORKS

NEURAL UNITS



1) Compute the neuron's activation

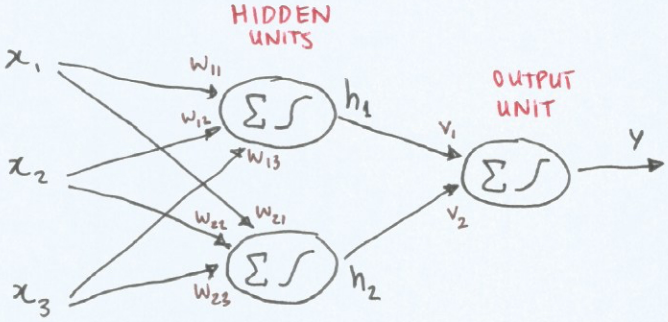
$$a = x^T w + w_0$$
$$= \sum_{d=1}^{D} x_d w_d + w_0$$

2) Set output as a $f\underline{n}$ of its activation $y = g(a)$

$$g(a) = \sigma(a) = \frac{1}{1 + e^{-a}}$$ i.e. sigmoid

3) If $y > 0.5$, assign x to class 1. Otherwise 0.

NEURAL NETWORK



Each unit also gets a bias weight!

To compute a class label in this network:

1) $h_1 \leftarrow g(w_1^T x + w_{10})$
2) $h_2 \leftarrow g(w_2^T x + w_{20})$
3) $y \leftarrow g\left(v^T \begin{pmatrix} h_1 \\ h_2 \end{pmatrix} + v_0\right)$
4) If $y > 0.5$, assign class 1.

→ To do regression instead of classification, don't squash the output.

Replace <u>step 3</u> w/ $\boxed{y \leftarrow g_3\left(v^T \begin{pmatrix} h_1 \\ h_2 \end{pmatrix} + v_0\right) \text{ where } g_3(a) = a}$

Identity $f\underline{n}$

MULTICLASS PREDICTION

→ Define one output for each class. (i.e. 3 classes → 3 output units)

→ <u>Step 3</u> is now:

$$\boxed{\forall m \in 1, 2, \ldots M, \quad y_m \leftarrow v_m^T \begin{pmatrix} h_1 \\ h_2 \end{pmatrix} + v_{m0}}$$

<u>Step 4</u> is now:

Prediction $f(x)$ is the class w/ the highest probability

$$f(x) = \max_{m=1}^{M} p(y = m \mid x)$$

SOFTMAX <u>FN</u>

(Same trick w/ Log Regr)

$$p(y = m \mid x) = \frac{e^{y_m}}{\sum_{k=1}^{M} e^{y_k}}$$

TRAINING ANNs

→ We want to find best weights for each unit

→ Create an error fn tht. measures the agreement of target y_i and the prediction $f(x)$

CLASSIFICATION

$$E = \sum_{i=1}^{n} y_i \log f(x_i) + (1-y_i) \log (1 - f(x_i))$$

REGRESSION

$$E = \sum_{i=1}^{n} (y_i - f(x_i))^2$$

→ It can make sense to use regularization penalty (e.g. $\lambda |w|^2$) to help control overfitting

↳ Called 'WEIGHT DECAY'

→ Find w so that E is minimized.

→ For Linear Regr. & Log. Regr., the optimization problem for w had a unique optimum.

For ANNs, there are local minima

→ We need the gradient descent of E w.r.t. all the params. w $\left[\text{i.e. } g(w) = \frac{\partial E}{\partial w} \right]$

(look at Optimization notes)

HOW TO DEAL W/ LOCAL MINIMA?

- Train multiple nets frm. diff. starting places and then choose best