Deep Learning
Summary of lecture 3

Dr. Richard E. Turner (ret26@cam.ac.uk)

Engineering Tripos Part IB
Paper 8: Information Engineering

Summary of lecture 3

1. relationship to maximum likelihood fit

$$G(\boldsymbol{w}) = -\sum_n \left[y^{(n)} \log \mathrm{x}(\boldsymbol{z}^{(n)};\boldsymbol{w}) + (1 - y^{(n)}) \log \left(1 - \mathrm{x}(\boldsymbol{z}^{(n)};\boldsymbol{w})\right)\right]$$

relative entropy / data fit

$$= -\, p(\{y^{(n)}\}_{n=1}^{N} | \{\boldsymbol{z}^{(n)}\}_{n=1}^{N}, \boldsymbol{w}) = -\prod_{n=1}^{N} p(y^{(n)}|\boldsymbol{z}^{(n)}, \boldsymbol{w})$$

negative log-likelihood of the parameters (Bernoulli dist.)

Summary of lecture 3

1. relationship to maximum likelihood fit

$$G(\boldsymbol{w}) = -\sum_n \left[y^{(n)} \log \mathrm{x}(\boldsymbol{z}^{(n)};\boldsymbol{w}) + (1 - y^{(n)}) \log\left(1 - \mathrm{x}(\boldsymbol{z}^{(n)};\boldsymbol{w})\right)\right]$$

relative entropy / data fit

$$= -\, p(\{y^{(n)}\}_{n=1}^{N} | \{\boldsymbol{z}^{(n)}\}_{n=1}^{N}, \boldsymbol{w}) = -\prod_{n=1}^{N} p(y^{(n)}|\boldsymbol{z}^{(n)}, \boldsymbol{w})$$

negative log-likelihood of the parameters (Bernoulli dist.)

2. the softmax function handles multiple classes e.g. $K = 3$ classes

$$x_1(\mathbf{z}; \{\mathbf{w}\}_{j=1}^{3}) = \frac{\exp(\mathbf{w}_1^\top \mathbf{z})}{\sum_{j=1}^{3} \exp(\mathbf{w}_j^\top \mathbf{z})} = p(y = 1|\{\mathbf{w}\}_{j=1}^{3}, \mathbf{z})$$

$$x_2(\mathbf{z}; \{\mathbf{w}\}_{j=1}^{3}) = \frac{\exp(\mathbf{w}_2^\top \mathbf{z})}{\sum_{j=1}^{3} \exp(\mathbf{w}_j^\top \mathbf{z})} = p(y = 2|\{\mathbf{w}\}_{j=1}^{3}, \mathbf{z})$$

$$x_3(\mathbf{z}; \{\mathbf{w}\}_{j=1}^{3}) = \frac{\exp(\mathbf{w}_3^\top \mathbf{z})}{\sum_{j=1}^{3} \exp(\mathbf{w}_j^\top \mathbf{z})} = p(y = 3|\{\mathbf{w}\}_{j=1}^{3}, \mathbf{z})$$

has a weight vector for each class

Summary of lecture 3

1. relationship to maximum likelihood fit

$$G(\boldsymbol{w}) = -\sum_n \left[y^{(n)} \log \mathrm{x}(\boldsymbol{z}^{(n)}; \boldsymbol{w}) + (1 - y^{(n)}) \log \left(1 - \mathrm{x}(\boldsymbol{z}^{(n)}; \boldsymbol{w})\right) \right]$$

relative entropy / data fit

$$= -\; p(\{y^{(n)}\}_{n=1}^N | \{\boldsymbol{z}^{(n)}\}_{n=1}^N, \boldsymbol{w}) = -\prod_{n=1}^{N} p(y^{(n)} | \boldsymbol{z}^{(n)}, \boldsymbol{w})$$

negative log-likelihood of the parameters (Bernoulli dist.)

2. the softmax function handles multiple classes e.g. $K = 3$ classes

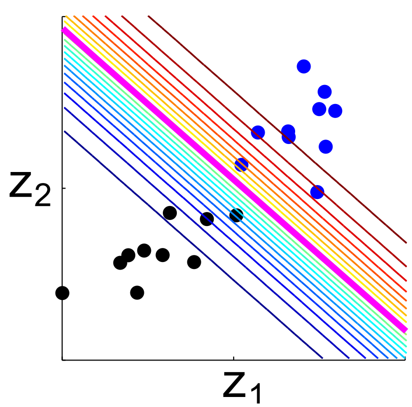
$$x_1(\mathbf{z}; \{\mathbf{w}\}_{j=1}^3) = \frac{\exp(\mathbf{w}_1^\top \mathbf{z})}{\sum_{j=1}^3 \exp(\mathbf{w}_j^\top \mathbf{z})} = p(y = 1 | \{\mathbf{w}\}_{j=1}^3, \mathbf{z})$$

$$x_2(\mathbf{z}; \{\mathbf{w}\}_{j=1}^3) = \frac{\exp(\mathbf{w}_2^\top \mathbf{z})}{\sum_{j=1}^3 \exp(\mathbf{w}_j^\top \mathbf{z})} = p(y = 2 | \{\mathbf{w}\}_{j=1}^3, \mathbf{z})$$

$$x_3(\mathbf{z}; \{\mathbf{w}\}_{j=1}^3) = \frac{\exp(\mathbf{w}_3^\top \mathbf{z})}{\sum_{j=1}^3 \exp(\mathbf{w}_j^\top \mathbf{z})} = p(y = 3 | \{\mathbf{w}\}_{j=1}^3, \mathbf{z})$$

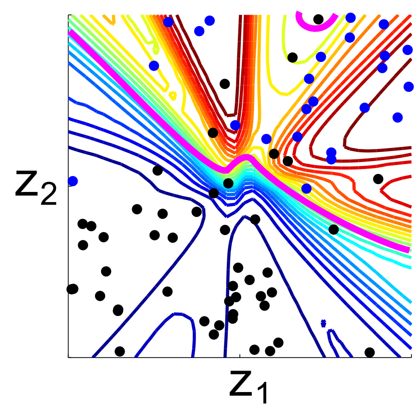
has a weight vector for each class

3. a single hidden layer neural network can make non-linear predictions



single neuron

linear decision boundary



neural network

non-linear decision boundary