Lecture 5

Cezar Ionescu

15/06/2019

DEPARTMENT FOR CONTINUING EDUCATION

UNIVERSITY OF OXFORD

Administrative

- Homework from 08/06/2019 due now!
- Please complete and hand in the declarations of authorship.

Questions?

McCulloch & Pitts

- McCulloch & Pitts 1943 *A Logical Calculus of the Ideas Immanent in Nervous Activity*

```
"neural events and the relations among them can be treated by
```

- The MC & P neuron had a number of boolean inputs, some positive and others negative. The neuron was activated if the number of active positive inputs was greater than the number of active negative inputs plus a “threshold”:

```
mc_p_neuron : ℝ -> ({0, 1}ⁿ, {0, 1}ᵐ) -> {0, 1}
mc_p_neuron θ (pos, neg) = if sum(pos) - sum(neg) ≥ θ
                           then 1 else 0
```

McCulloch & Pitts

Logical functions:

```
not : {0, 1} -> {0, 1}
not x = mc_p_neuron (-0.5) ([], [x])

and : ({0, 1}, {0, 1}) -> {0, 1}
and (x, y) = mc_p_neuron (-1.5) ([x, y],[])
```

Perceptrons

- Frank Rosenblatt 1957
- An FR neuron had real-valued inputs and binary outputs. The output was a step function of the weighted sum of these inputs

```
fr_neuron : (ℝⁿ, ℝ) -> {0, 1}ⁿ -> {0, 1}
fr_neuron ([w₁,..., wₙ], θ) [x₁, ..., xₙ] = if s ≥ θ then 1
                                                      else -1
 where s = w₁*x₁ + ... + wₙ * xₙ
```

Perceptrons

Logical functions:

Perceptrons

The perceptron training rule:

```
wᵢ <- wᵢ + η * (t - o) * xᵢ
```

Perceptrons

The case of `xor`:

Perceptrons

Linear separability

Perceptrons

Implementing `xor`

Gradient descent

Why does the perceptron training rule work?

"Naive" gradient descent:

```
x <- x - η * D f (x)
```

What is `f` in our case?

Error functions

```
t - o, |t - o|, (t - o)², (t - o)⁴, ...
```

Bayesian learning and error minimisation

```
data: (x₁, t₁), ..., (xₙ, tₙ), xᵢ ∈ ℝᵐ, tᵢ ∈ ℝ
hypothesis: w ∈ ℝᵐ

hₘₐₚ = argmax p(h | d)
     = argmax p(d | h) * p(h) / p(d)
     = argmax p(d | h) * p(h)
     -- assume p(h) = const
     = argmax p(d | h)
     = hₘₗ
```

Bayesian learning and error minimisation

$$p(d \mid h) = p((x_1, t_1), \ldots, (x_n, t_n) \mid w)$$
$$= p(x_1, t_1 \mid w) * \ldots * p(x_n, t_n \mid w)$$

$$h_{ml} = \operatorname{argmax}\ p(d \mid h)$$
$$= \operatorname{argmax}\ \ln p(d \mid h)$$
$$= \operatorname{argmax}\ \ln p(x_1, t_1 \mid w) + \ldots + \ln p(x_n, t_n \mid w)$$

Bayesian learning and error minimisation

Assume the $f(x_i, w)$ are normally distributed around the t_i, with the **same** σ:

$$p(x_i, t_i \mid w) = 1/\sqrt{(2 * \pi * \sigma^2)} \exp(-(t_i - f(x_i, w))^2/(2 * \sigma^2))$$

Therefore

$$\begin{aligned} \ln p(x_i, t_i \mid w) &= \ln 1/\sqrt{(2 * \pi * \sigma^2)} - (t_i - f(x_i, w))^2/(2 * \sigma^2) \\ &= k - (t_i - f(x_i, w))^2/(2 * \sigma^2) \end{aligned}$$

Bayesian learning and error minimisation

$$
\begin{aligned}
h_{ml} &= \text{argmax } \ln p(x_1, t_1 \mid w) + \ldots + \ln p(x_n, t_n \mid w) \\
&= \text{argmax } n * k - \Sigma\, (t_i - f(x_i, w))^2 / (2 * \sigma\char`^2) \\
&= \text{argmax } - \Sigma\, (t_i - f(x_i, w))^2 \\
&= \text{argmin } \Sigma\, (t_i - f(x_i, w))^2
\end{aligned}
$$

Bayesian learning and error minimisation

Therefore, the correct error to choose is the sum of squared errors…

…at least if:

- all hypothesis are equally likely a-priori,
- the errors are independent, and
- the errors have identical normal distributions.