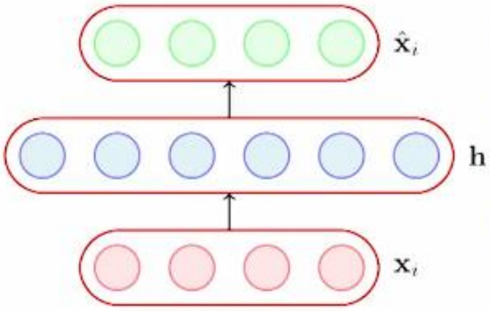
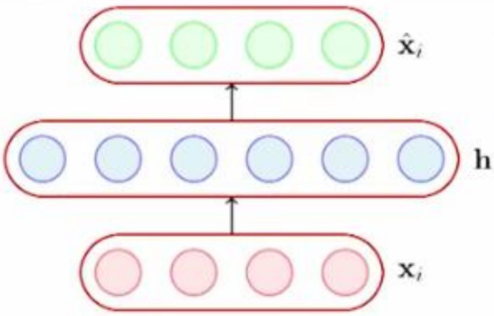


- A hidden neuron with sigmoid activation will have values between 0 and 1
- We say that the neuron is activated when its output is close to 1 and not activated when its output is close to 0.
- A sparse autoencoder tries to ensure the neuron is inactive most of the times.

0:58 / 9:11



- If the neuron l is sparse (i.e. mostly inactive) then $\hat{\rho}_l \to 0$
- A sparse autoencoder uses a sparsity parameter ρ (typically very close to 0, say, 0.005) and tries to enforce the constraint $\hat{\rho}_l = \rho$
- One way of ensuring this is to add the following term to the objective function

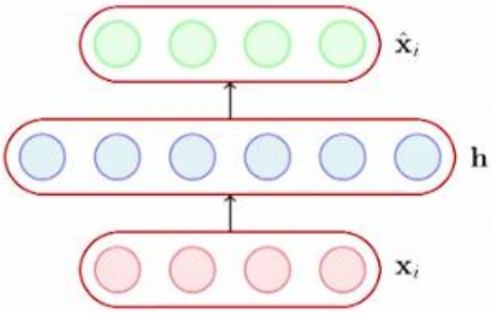
The average value of the activation of a neuron l is given by

$$\hat{\rho}_l = \frac{1}{m}\sum_{i=1}^{m} h(\mathbf{x}_i)_l$$

$$\Omega(\theta) = \sum_{l=1}^{k} \rho \log \frac{\rho}{\hat{\rho}_l} + (1-\rho)\log\frac{1-\rho}{1-\hat{\rho}_l}$$

$$\mathcal{L}(\theta) = \underline{\underline{\mathcal{L}'(\theta)}} + \underline{\underline{\Omega(\theta)}}$$

4:44 / 9:11



- If the neuron l is sparse (i.e. mostly inactive) then $\hat{\rho}_l \to 0$
- A sparse autoencoder uses a sparsity parameter ρ (typically very close to 0, say, 0.005) and tries to enforce the constraint $\hat{\rho}_l = \rho$
- One way of ensuring this is to add the following term to the objective function

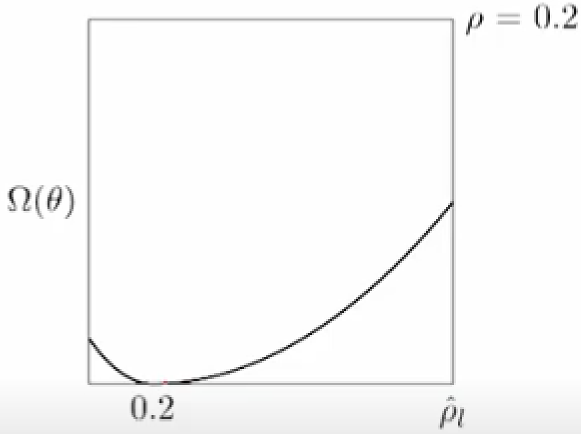
The average value of the activation of a neuron l is given by

$$\hat{\rho}_l = \frac{1}{m}\sum_{i=1}^{m} h(\mathbf{x}_i)_l$$

$$\Omega(\theta) = \sum_{l=1}^{k} \rho \log \frac{\rho}{\hat{\rho}_l} + (1-\rho)\log \frac{1-\rho}{1-\hat{\rho}_l}$$

- When will this term reach its minimum value and what is the minimum value? Let us plot it and check.

5:23 / 9:11



- The function will reach its minimum value(s) when $\hat{\rho}_l = \rho$.

$$\Omega(\theta) = \sum_{l=1}^{k} \rho log \frac{\rho}{\hat{\rho}_l} + (1-\rho) log \frac{1-\rho}{1-\hat{\rho}_l}$$

Can be re-written as

$$-\Omega(\theta) = \sum_{l=1}^{k} \rho log \rho - \rho log \hat{\rho}_l + (1-\rho) log(1-\rho) - (1-\rho) log(1-\hat{\rho}_l)$$

By Chain rule

$$\frac{\partial \Omega(\theta)}{\partial W} = \frac{\partial \Omega(\theta)}{\partial \hat{\rho}} * \frac{\partial \hat{\rho}}{\partial W}$$

$$\frac{\partial \Omega(\theta)}{\partial \hat{\rho}} = -\frac{\rho}{\hat{\rho}} + \frac{(1-\rho)}{1-\hat{\rho}}$$

For each neuron $l \in 1 \ldots k$ in hidden layer, we have

$$\frac{\partial \hat{\rho}_l}{\partial W} = \mathbf{x}_i (g'(W^T \mathbf{x}_i + \mathbf{b}))^T$$

Finally,

$$\frac{\partial \hat{\mathscr{L}}(\theta)}{\partial W} = \frac{\partial \mathscr{L}(\theta)}{\partial W} + \frac{\partial \Omega(\theta)}{\partial W}$$

(and we know how to calculate both terms on R.H.S)

- Now,

$$\hat{\mathscr{L}}(\theta) = \mathscr{L}(\theta) + \Omega(\theta)$$

- $\mathscr{L}(\theta)$ is the squared error loss or cross entropy loss and $\Omega(\theta)$ is the sparsity constraint.
- We already know how to calculate $\frac{\partial \mathscr{L}(\theta)}{\partial W}$
- Let us see how to calculate $\frac{\partial \Omega(\theta)}{\partial W}$.

8:01 / 9:11

Derivation

$$\frac{\partial \hat{\rho}}{\partial W} = \left[\frac{\partial \hat{\rho}_1}{\partial W} \quad \frac{\partial \hat{\rho}_2}{\partial W} \cdots \frac{\partial \hat{\rho}_k}{\partial W}\right]$$

For each element in the above equation we can calculate $\frac{\partial \hat{\rho}_l}{\partial W}$ (which is the partial derivative of a scalar w.r.t. a matrix = matrix). For a single element of a matrix W_{jl}:-

$$\begin{aligned}
\frac{\partial \hat{\rho}_l}{\partial W_{jl}} &= \frac{\partial\left[\frac{1}{m}\sum_{i=1}^{m} g(W_{:,l}^T \mathbf{x_i} + b_l)\right]}{\partial W_{jl}} \\
&= \frac{1}{m}\sum_{i=1}^{m} \frac{\partial\left[g(W_{:,l}^T \mathbf{x_i} + b_l)\right]}{\partial W_{jl}} \\
&= \frac{1}{m}\sum_{i=1}^{m} g'(W_{:,l}^T \mathbf{x_i} + b_l) x_{ij}
\end{aligned}$$

So in matrix notation we can write it as :

$$\frac{\partial \hat{\rho}_l}{\partial W} = \mathbf{x}_i (g'(W^T \mathbf{x}_i + \mathbf{b}))^T$$

8:07 / 9:11