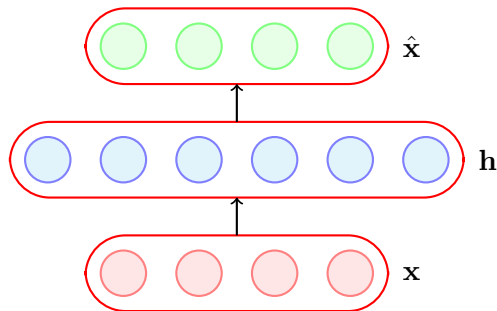


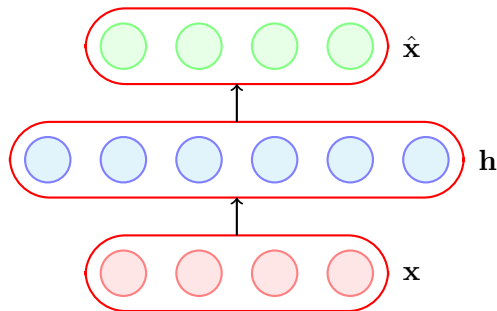
Module 7.6: Contractive Autoencoders

- A contractive autoencoder also tries to prevent an overcomplete autoencoder from learning the identity function.



- A contractive autoencoder also tries to prevent an overcomplete autoencoder from learning the identity function.
- It does so by adding the following regularization term to the loss function

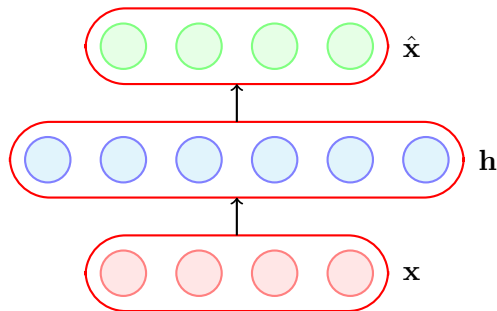
$$\Omega(\theta) = \|J_{\mathbf{x}}(\mathbf{h})\|_F^2$$



- A contractive autoencoder also tries to prevent an overcomplete autoencoder from learning the identity function.
- It does so by adding the following regularization term to the loss function

$$\Omega(\theta) = \|J_{\mathbf{x}}(\mathbf{h})\|_F^2$$

where $J_{\mathbf{x}}(\mathbf{h})$ is the Jacobian of the encoder.

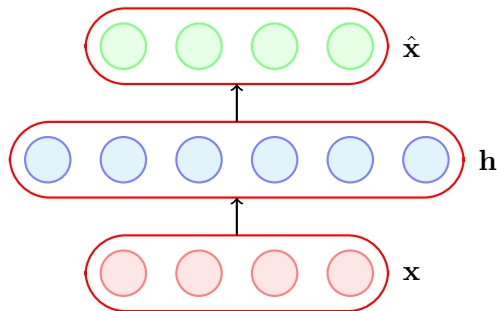


- A contractive autoencoder also tries to prevent an overcomplete autoencoder from learning the identity function.
- It does so by adding the following regularization term to the loss function

$$\Omega(\theta) = \|J_{\mathbf{x}}(\mathbf{h})\|_F^2$$

where $J_{\mathbf{x}}(\mathbf{h})$ is the Jacobian of the encoder.

- Let us see what it looks like.



- If the input has n dimensions and the hidden layer has k dimensions then

- If the input has n dimensions and the hidden layer has k dimensions then

$$J_{\mathbf{x}}(\mathbf{h}) = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \cdots & \cdots & \cdots & \frac{\partial h_1}{\partial x_n} \\ \frac{\partial h_2}{\partial x_1} & \cdots & \cdots & \cdots & \frac{\partial h_2}{\partial x_n} \\ \vdots & & \ddots & & \vdots \\ \frac{\partial h_k}{\partial x_1} & \cdots & \cdots & \cdots & \frac{\partial h_k}{\partial x_n} \end{bmatrix}$$

- If the input has n dimensions and the hidden layer has k dimensions then
- In other words, the (j, l) entry of the Jacobian captures the variation in the output of the l^{th} neuron with a small variation in the j^{th} input.

$$J_{\mathbf{x}}(\mathbf{h}) = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \cdots & \cdots & \cdots & \frac{\partial h_1}{\partial x_n} \\ \frac{\partial h_2}{\partial x_1} & \cdots & \cdots & \cdots & \frac{\partial h_2}{\partial x_n} \\ \vdots & & \ddots & & \vdots \\ \frac{\partial h_k}{\partial x_1} & \cdots & \cdots & \cdots & \frac{\partial h_k}{\partial x_n} \end{bmatrix}$$

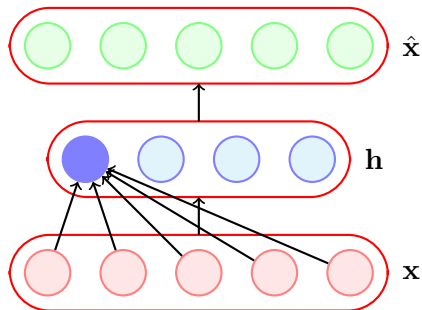
- If the input has n dimensions and the hidden layer has k dimensions then
- In other words, the (j, l) entry of the Jacobian captures the variation in the output of the l^{th} neuron with a small variation in the j^{th} input.

$$J_{\mathbf{x}}(\mathbf{h}) = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \cdots & \cdots & \cdots & \frac{\partial h_1}{\partial x_n} \\ \frac{\partial h_2}{\partial x_1} & \cdots & \cdots & \cdots & \frac{\partial h_2}{\partial x_n} \\ \vdots & & \ddots & & \vdots \\ \frac{\partial h_k}{\partial x_1} & \cdots & \cdots & \cdots & \frac{\partial h_k}{\partial x_n} \end{bmatrix}$$

$$\|J_{\mathbf{x}}(\mathbf{h})\|_F^2 = \sum_{j=1}^n \sum_{l=1}^k \left(\frac{\partial h_l}{\partial x_j} \right)^2$$

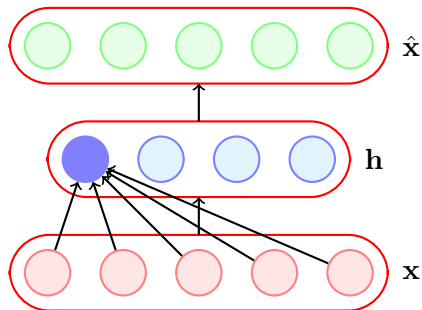
- What is the intuition behind this ?

$$\|J_{\mathbf{x}}(\mathbf{h})\|_F^2 = \sum_{j=1}^n \sum_{l=1}^k \left(\frac{\partial h_l}{\partial x_j} \right)^2$$



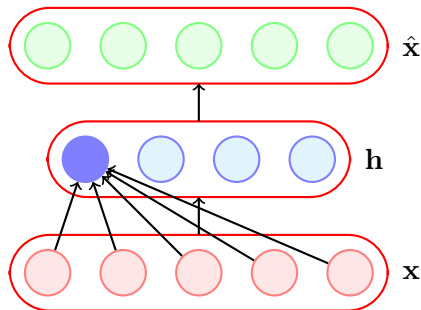
- What is the intuition behind this ?
- Consider $\frac{\partial h_1}{\partial x_1}$, what does it mean if $\frac{\partial h_1}{\partial x_1} = 0$

$$\|J_{\mathbf{x}}(\mathbf{h})\|_F^2 = \sum_{j=1}^n \sum_{l=1}^k \left(\frac{\partial h_l}{\partial x_j} \right)^2$$



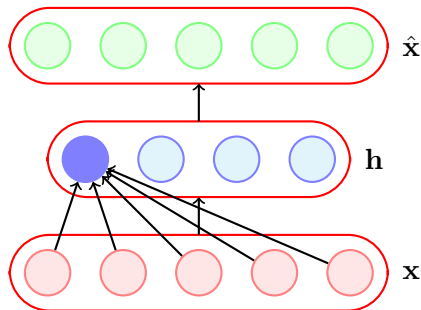
- What is the intuition behind this ?
- Consider $\frac{\partial h_1}{\partial x_1}$, what does it mean if $\frac{\partial h_1}{\partial x_1} = 0$
- It means that this neuron is not very sensitive to variations in the input x_1 .

$$\|J_{\mathbf{x}}(\mathbf{h})\|_F^2 = \sum_{j=1}^n \sum_{l=1}^k \left(\frac{\partial h_l}{\partial x_j} \right)^2$$



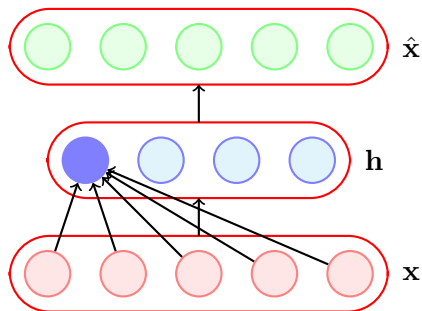
- What is the intuition behind this ?
- Consider $\frac{\partial h_1}{\partial x_1}$, what does it mean if $\frac{\partial h_1}{\partial x_1} = 0$
- It means that this neuron is not very sensitive to variations in the input x_1 .
- But doesn't this contradict our other goal of minimizing $\mathcal{L}(\theta)$ which requires \mathbf{h} to capture variations in the input.

$$\|J_{\mathbf{x}}(\mathbf{h})\|_F^2 = \sum_{j=1}^n \sum_{l=1}^k \left(\frac{\partial h_l}{\partial x_j} \right)^2$$



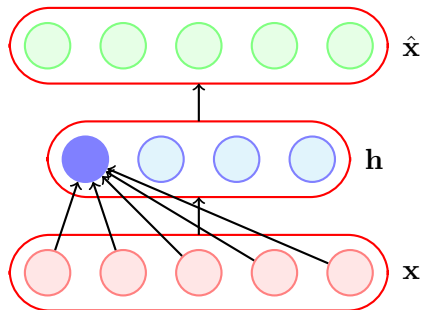
- Indeed it does and that's the idea

$$\|J_{\mathbf{x}}(\mathbf{h})\|_F^2 = \sum_{j=1}^n \sum_{l=1}^k \left(\frac{\partial h_l}{\partial x_j} \right)^2$$



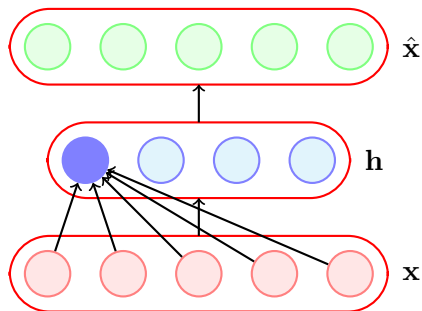
- Indeed it does and that's the idea
- By putting these two contradicting objectives against each other we ensure that \mathbf{h} is sensitive to only very important variations as observed in the training data.

$$\|J_{\mathbf{x}}(\mathbf{h})\|_F^2 = \sum_{j=1}^n \sum_{l=1}^k \left(\frac{\partial h_l}{\partial x_j} \right)^2$$



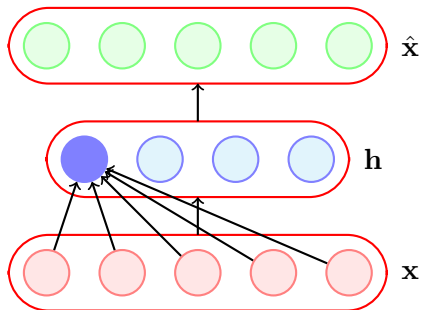
- Indeed it does and that's the idea
- By putting these two contradicting objectives against each other we ensure that \mathbf{h} is sensitive to only very important variations as observed in the training data.
- $\mathcal{L}(\theta)$ - capture important variations in data

$$\|J_{\mathbf{x}}(\mathbf{h})\|_F^2 = \sum_{j=1}^n \sum_{l=1}^k \left(\frac{\partial h_l}{\partial x_j} \right)^2$$



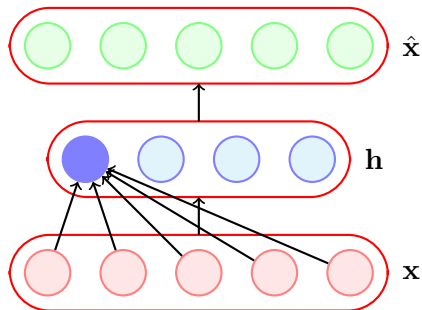
- Indeed it does and that's the idea
- By putting these two contradicting objectives against each other we ensure that \mathbf{h} is sensitive to only very important variations as observed in the training data.
- $\mathcal{L}(\theta)$ - capture important variations in data
- $\Omega(\theta)$ - do not capture variations in data

$$\|J_{\mathbf{x}}(\mathbf{h})\|_F^2 = \sum_{j=1}^n \sum_{l=1}^k \left(\frac{\partial h_l}{\partial x_j} \right)^2$$

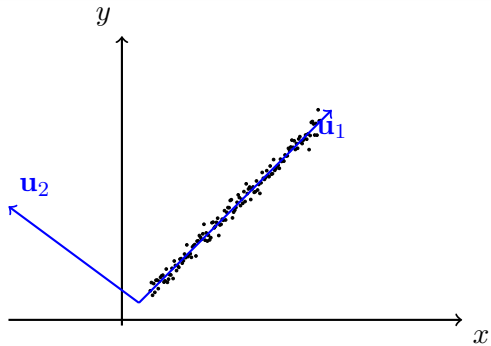


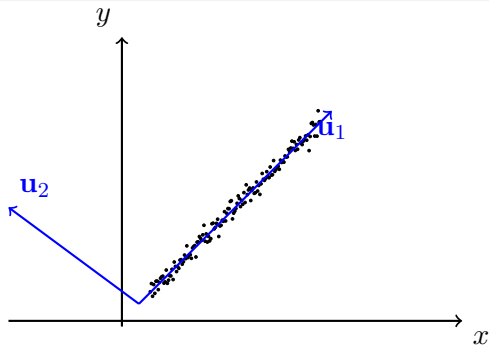
- Indeed it does and that's the idea
- By putting these two contradicting objectives against each other we ensure that \mathbf{h} is sensitive to only very important variations as observed in the training data.
- $\mathcal{L}(\theta)$ - capture important variations in data
- $\Omega(\theta)$ - do not capture variations in data
- Tradeoff - capture only very important variations in the data

$$\|J_{\mathbf{x}}(\mathbf{h})\|_F^2 = \sum_{j=1}^n \sum_{l=1}^k \left(\frac{\partial h_l}{\partial x_j} \right)^2$$

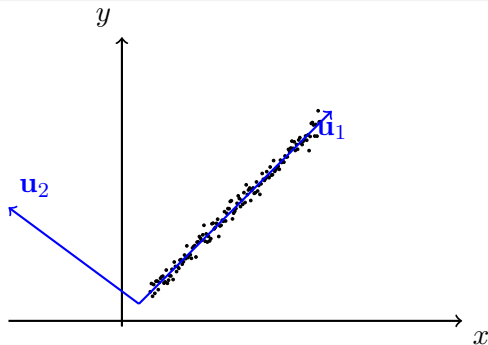


Let us try to understand this with the help of an illustration.

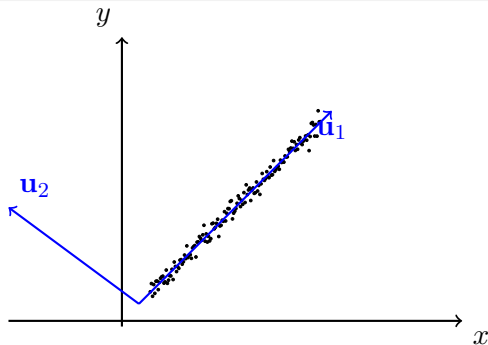




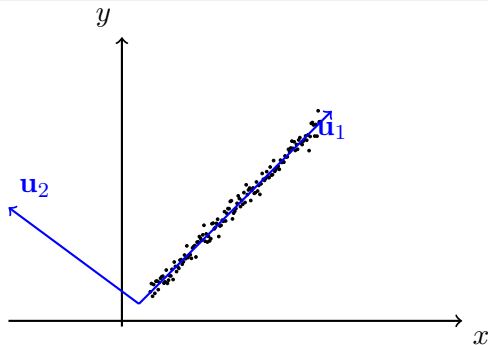
- Consider the variations in the data along directions \mathbf{u}_1 and \mathbf{u}_2



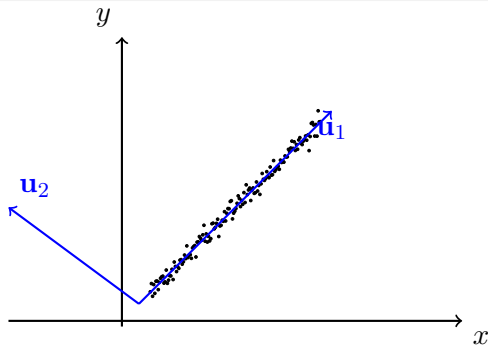
- Consider the variations in the data along directions \mathbf{u}_1 and \mathbf{u}_2
- It makes sense to maximize a neuron to be sensitive to variations along \mathbf{u}_1



- Consider the variations in the data along directions \mathbf{u}_1 and \mathbf{u}_2
- It makes sense to maximize a neuron to be sensitive to variations along \mathbf{u}_1
- At the same time it makes sense to inhibit a neuron from being sensitive to variations along \mathbf{u}_2 (as there seems to be small noise and unimportant for reconstruction)



- Consider the variations in the data along directions \mathbf{u}_1 and \mathbf{u}_2
- It makes sense to maximize a neuron to be sensitive to variations along \mathbf{u}_1
- At the same time it makes sense to inhibit a neuron from being sensitive to variations along \mathbf{u}_2 (as there seems to be small noise and unimportant for reconstruction)
- By doing so we can balance between the contradicting goals of good reconstruction and low sensitivity.



- Consider the variations in the data along directions \mathbf{u}_1 and \mathbf{u}_2
- It makes sense to maximize a neuron to be sensitive to variations along \mathbf{u}_1
- At the same time it makes sense to inhibit a neuron from being sensitive to variations along \mathbf{u}_2 (as there seems to be small noise and unimportant for reconstruction)
- By doing so we can balance between the contradicting goals of good reconstruction and low sensitivity.
- What does this remind you of ?