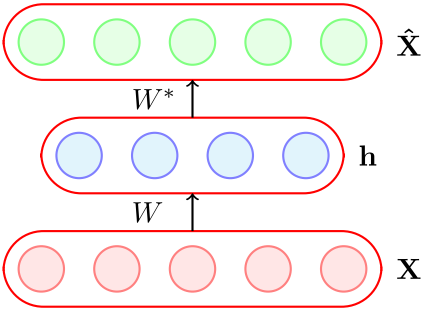
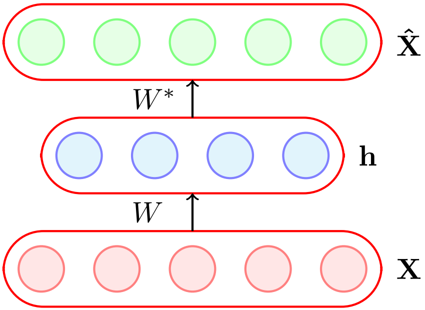
Module 21.1: Revisiting Autoencoders



$$\mathbf{h} = g(W\mathbf{X} + \mathbf{b})$$

$$\hat{\mathbf{X}} = f(W^*\mathbf{h} + \mathbf{c})$$

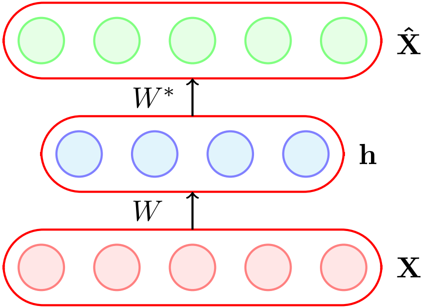
- Before we start talking about VAEs, let us quickly revisit autoencoders



$$\mathbf{h} = g(W\mathbf{X} + \mathbf{b})$$

$$\hat{\mathbf{X}} = f(W^*\mathbf{h} + \mathbf{c})$$

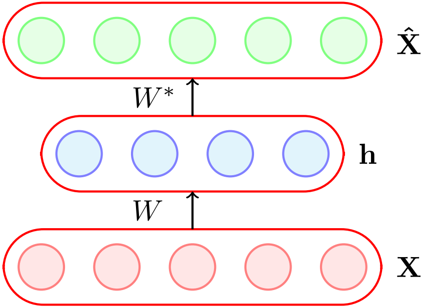
- Before we start talking about VAEs, let us quickly revisit autoencoders
- An autoencoder contains an encoder which takes the input X and maps it to a hidden representation



$$\mathbf{h} = g(W\mathbf{X} + \mathbf{b})$$

$$\hat{\mathbf{X}} = f(W^*\mathbf{h} + \mathbf{c})$$

- Before we start talking about VAEs, let us quickly revisit autoencoders
- An autoencoder contains an encoder which takes the input X and maps it to a hidden representation
- The decoder then takes this hidden representation and tries to reconstruct the input from it as $\hat{X}$

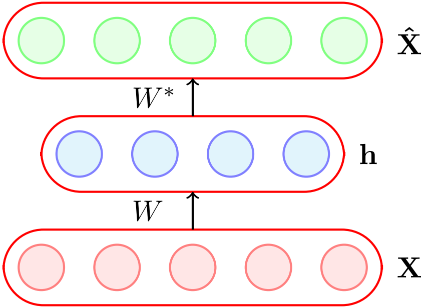


$$\mathbf{h} = g(W\mathbf{X} + \mathbf{b})$$

$$\hat{\mathbf{X}} = f(W^*\mathbf{h} + \mathbf{c})$$

- Before we start talking about VAEs, let us quickly revisit autoencoders
- An autoencoder contains an encoder which takes the input X and maps it to a hidden representation
- The decoder then takes this hidden representation and tries to reconstruct the input from it as $\hat{X}$
- The training happens using the following objective function

$$\min_{W,W^*,\mathbf{c},\mathbf{b}} \frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{n} (\hat{x}_{ij} - x_{ij})^2$$



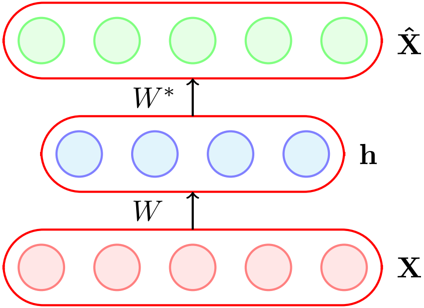
$$\mathbf{h} = g(W\mathbf{X} + \mathbf{b})$$

$$\hat{\mathbf{X}} = f(W^*\mathbf{h} + \mathbf{c})$$

- Before we start talking about VAEs, let us quickly revisit autoencoders
- An autoencoder contains an encoder which takes the input X and maps it to a hidden representation
- The decoder then takes this hidden representation and tries to reconstruct the input from it as $\hat{X}$
- The training happens using the following objective function

$$\min_{W,W^*,\mathbf{c},\mathbf{b}} \frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{n} (\hat{x}_{ij} - x_{ij})^2$$

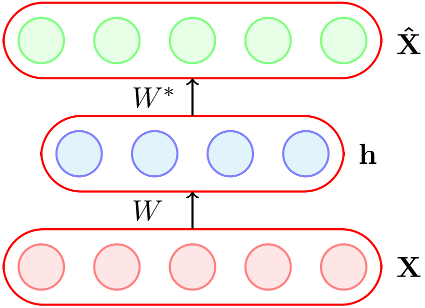
- where m is the number of training instances, $\{x_i\}_{i=1}^{m}$ and each $x_i \in R^n$ (x_{ij} is thus the j-th dimension of the i-th training instance)



$$\mathbf{h} = g(W\mathbf{X} + \mathbf{b})$$

$$\hat{\mathbf{X}} = f(W^*\mathbf{h} + \mathbf{c})$$

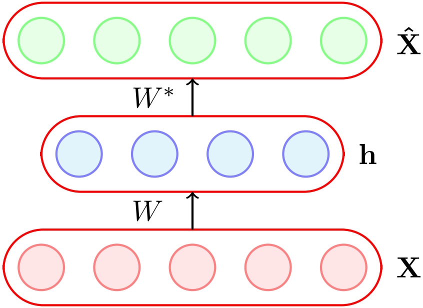
- But where's the fun in this ?



$$\mathbf{h} = g(W\mathbf{X} + \mathbf{b})$$

$$\hat{\mathbf{X}} = f(W^*\mathbf{h} + \mathbf{c})$$

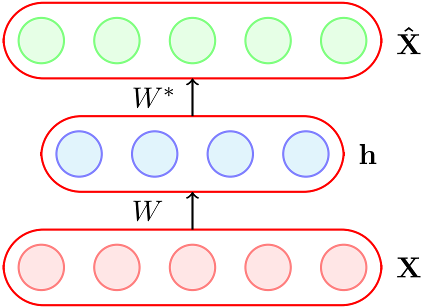
- But where's the fun in this ?
- We are taking an input and simply reconstructing it



$$\mathbf{h} = g(W\mathbf{X} + \mathbf{b})$$

$$\hat{\mathbf{X}} = f(W^*\mathbf{h} + \mathbf{c})$$

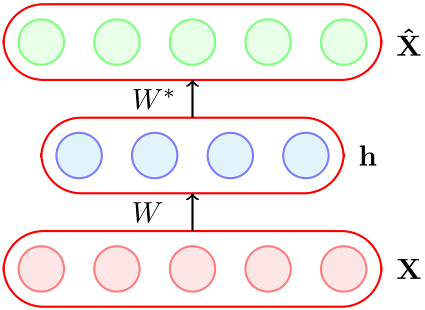
- But where's the fun in this ?
- We are taking an input and simply reconstructing it
- Of course, the fun lies in the fact that we are getting a good *abstraction* of the input



$$\mathbf{h} = g(W\mathbf{X} + \mathbf{b})$$

$$\hat{\mathbf{X}} = f(W^*\mathbf{h} + \mathbf{c})$$

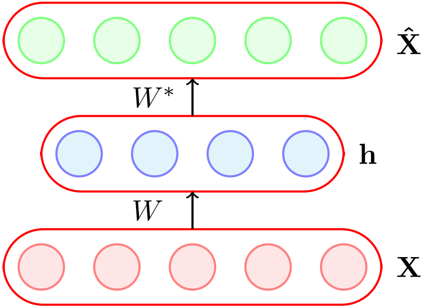
- But where's the fun in this ?
- We are taking an input and simply reconstructing it
- Of course, the fun lies in the fact that we are getting a good *abstraction* of the input
- But RBMs were able to do something more besides abstraction



$$\mathbf{h} = g(W\mathbf{X} + \mathbf{b})$$

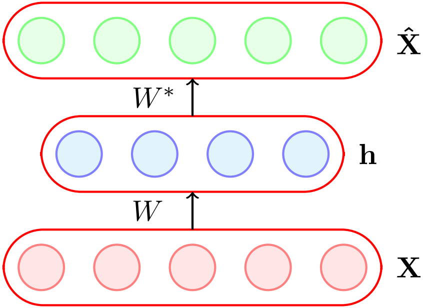
$$\hat{\mathbf{X}} = f(W^*\mathbf{h} + \mathbf{c})$$

- But where's the fun in this ?
- We are taking an input and simply reconstructing it
- Of course, the fun lies in the fact that we are getting a good *abstraction* of the input
- But RBMs were able to do something more besides abstraction (they were able to do *generation*)



$$\mathbf{h} = g(W\mathbf{X} + \mathbf{b})$$
$$\hat{\mathbf{X}} = f(W^*\mathbf{h} + \mathbf{c})$$

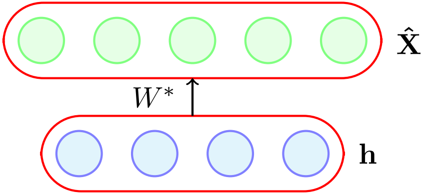
- But where's the fun in this ?
- We are taking an input and simply reconstructing it
- Of course, the fun lies in the fact that we are getting a good *abstraction* of the input
- But RBMs were able to do something more besides abstraction (they were able to do *generation*)
- Let us revisit *generation* in the context of autoencoders



$$\mathbf{h} = g(W\mathbf{X} + \mathbf{b})$$

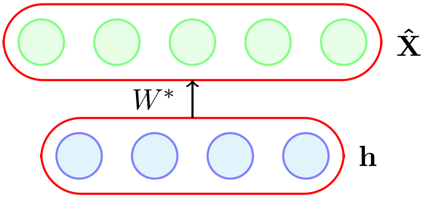
$$\hat{\mathbf{X}} = f(W^*\mathbf{h} + \mathbf{c})$$

- Can we do generation with autoencoders ?



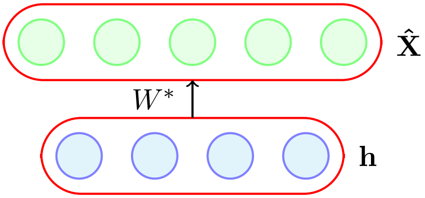
$$\hat{\mathbf{X}} = f(W^*\mathbf{h} + \mathbf{c})$$

- Can we do generation with autoencoders ?
- In other words, once the autoencoder is trained can I remove the encoder, feed a hidden representation h to the decoder and decode a $\hat{X}$ from it ?



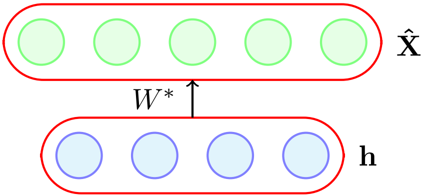
$$\hat{\mathbf{X}} = f(W^*\mathbf{h} + \mathbf{c})$$

- Can we do generation with autoencoders ?
- In other words, once the autoencoder is trained can I remove the encoder, feed a hidden representation h to the decoder and decode a $\hat{X}$ from it ?
- In principle, yes! But in practice there is a problem with this approach



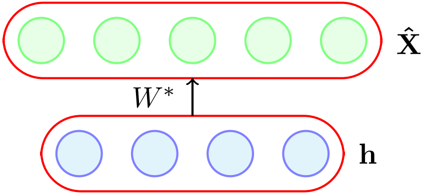
$$\hat{\mathbf{X}} = f(W^*\mathbf{h} + \mathbf{c})$$

- Can we do generation with autoencoders ?
- In other words, once the autoencoder is trained can I remove the encoder, feed a hidden representation h to the decoder and decode a $\hat{X}$ from it ?
- In principle, yes! But in practice there is a problem with this approach
- h is a very high dimensional vector and only a few vectors in this space would actually correspond to meaningful latent representations of our input



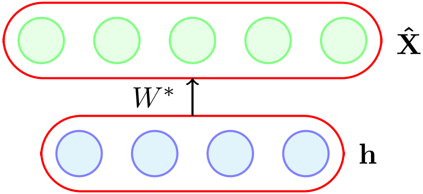
$$\hat{\mathbf{X}} = f(W^*\mathbf{h} + \mathbf{c})$$

- Can we do generation with autoencoders ?
- In other words, once the autoencoder is trained can I remove the encoder, feed a hidden representation h to the decoder and decode a $\hat{X}$ from it ?
- In principle, yes! But in practice there is a problem with this approach
- h is a very high dimensional vector and only a few vectors in this space would actually correspond to meaningful latent representations of our input
- So of all the possible value of h which values should I feed to the decoder (we had asked a similar question before: slide 67, bullet 5 of lecture 19)



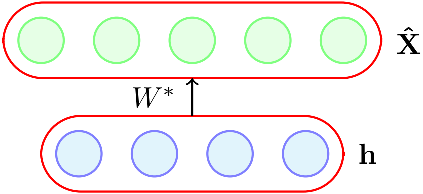
$$\hat{\mathbf{X}} = f(W^*\mathbf{h} + \mathbf{c})$$

- Ideally, we should only feed those values of h which are highly *likely*



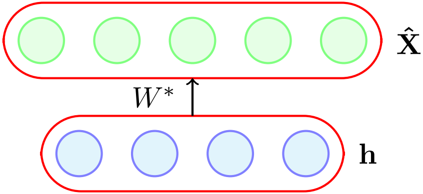
- Ideally, we should only feed those values of h which are highly *likely*
- In other words, we are interested in sampling from $P(h|X)$ so that we pick only those h's which have a high probability

$$\hat{\mathbf{X}} = f(W^*\mathbf{h} + \mathbf{c})$$



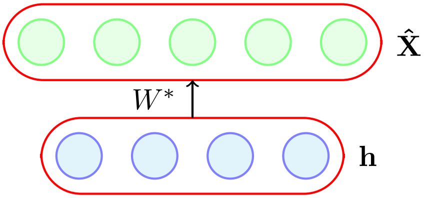
$$\hat{\mathbf{X}} = f(W^*\mathbf{h} + \mathbf{c})$$

- Ideally, we should only feed those values of h which are highly *likely*
- In other words, we are interested in sampling from $P(h|X)$ so that we pick only those h's which have a high probability
- But unlike RBMs, autoencoders do not have such a probabilistic interpretation



$$\hat{\mathbf{X}} = f(W^*\mathbf{h} + \mathbf{c})$$

- Ideally, we should only feed those values of h which are highly *likely*
- In other words, we are interested in sampling from $P(h|X)$ so that we pick only those h's which have a high probability
- But unlike RBMs, autoencoders do not have such a probabilistic interpretation
- They learn a hidden representation h but not a distribution $P(h|X)$



$$\hat{\mathbf{X}} = f(W^*\mathbf{h} + \mathbf{c})$$

- Ideally, we should only feed those values of h which are highly *likely*
- In other words, we are interested in sampling from $P(h|X)$ so that we pick only those h's which have a high probability
- But unlike RBMs, autoencoders do not have such a probabilistic interpretation
- They learn a hidden representation h but not a distribution $P(h|X)$
- Similarly the decoder is also deterministic and does not learn a distribution over X (given a h we can get a X but not $P(X|h)$)

We will now look at variational autoencoders which have the same structure as autoencoders but they learn a distribution over the hidden variables