

Variational Autoencoder (VAE)

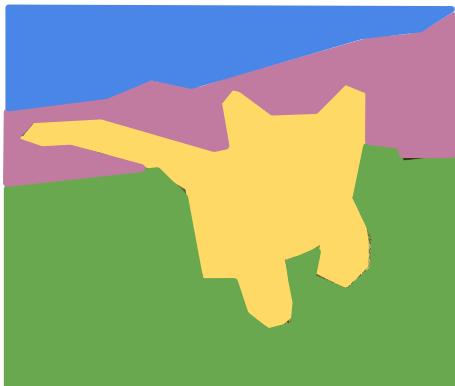
Supervised learning

Supervised learning



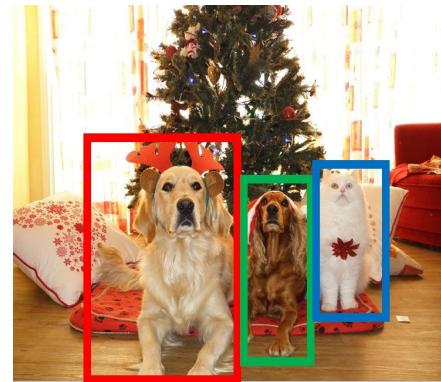
→ Cat

Classification



GRASS, CAT,
TREE, SKY

Semantic Segmentation



DOG, DOG, CAT

Object Detection

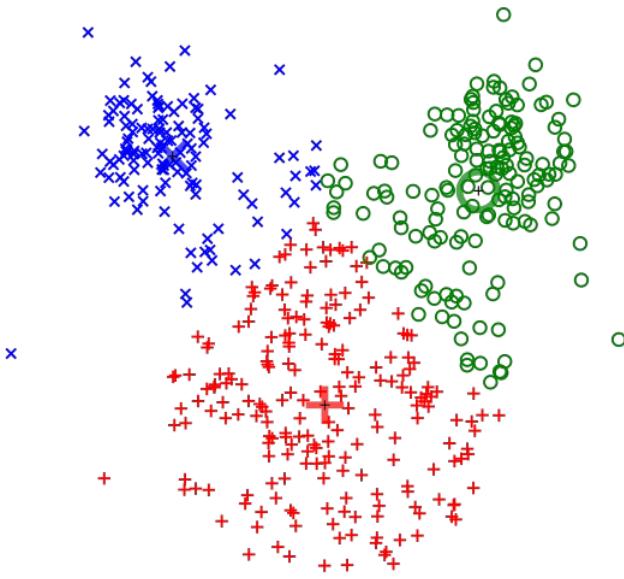


A cat sitting on a suitcase on the floor

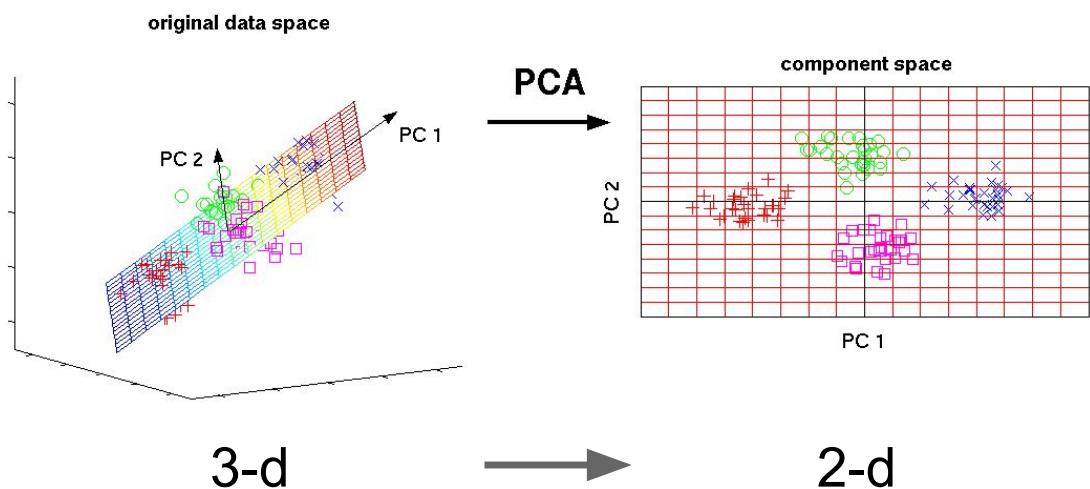
Image captioning

Unsupervised learning

Unsupervised learning

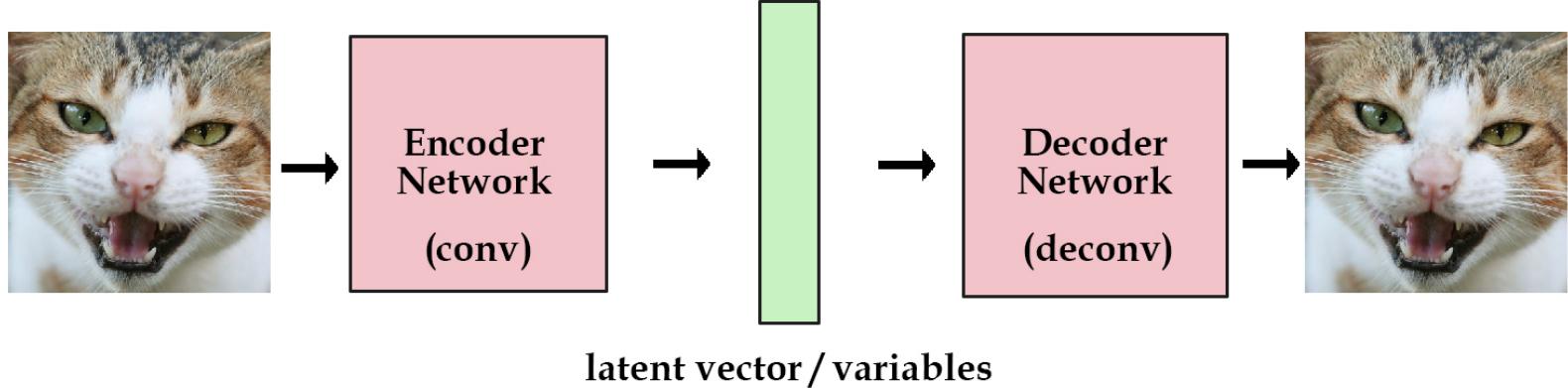


K-means clustering



Principal Component Analysis
(Dimensionality reduction)

Autoencoder



```
class autoencoder(nn.Module):
    def __init__(self):
        super(autoencoder, self).__init__()
        self.encoder = nn.Sequential(
            nn.Conv2d(1, 16, 3, stride=3, padding=1), # b, 16, 10, 10
            nn.ReLU(True),
            nn.MaxPool2d(2, stride=2), # b, 16, 5, 5
            nn.Conv2d(16, 8, 3, stride=2, padding=1), # b, 8, 3, 3
            nn.ReLU(True),
            nn.MaxPool2d(2, stride=1) # b, 8, 2, 2
        )
        self.decoder = nn.Sequential(
            nn.ConvTranspose2d(8, 16, 3, stride=2), # b, 16, 5, 5
            nn.ReLU(True),
            nn.ConvTranspose2d(16, 8, 5, stride=3, padding=1), # b, 8, 15, 15
            nn.ReLU(True),
            nn.ConvTranspose2d(8, 1, 2, stride=2, padding=1), # b, 1, 28, 28
            nn.Tanh()
        )

    def forward(self, x):
        x = self.encoder(x)
        x = self.decoder(x)
        return x
```

```
model = autoencoder().cuda()
criterion = nn.MSELoss()
optimizer = torch.optim.Adam(model.parameters(), lr=learning_rate,
                             weight_decay=1e-5)

for epoch in range(num_epochs):
    for data in dataloader:
        img, _ = data
        img = Variable(img).cuda()
        # =====forward=====
        output = model(img)
        loss = criterion(output, img)
        # =====backward=====
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
```

Transposed convolution

A guide to convolution arithmetic for deep
learning

Vincent Dumoulin¹★ and Francesco Visin²★†

★MILA, Université de Montréal

†AIRLab, Politecnico di Milano

Transposed convolution

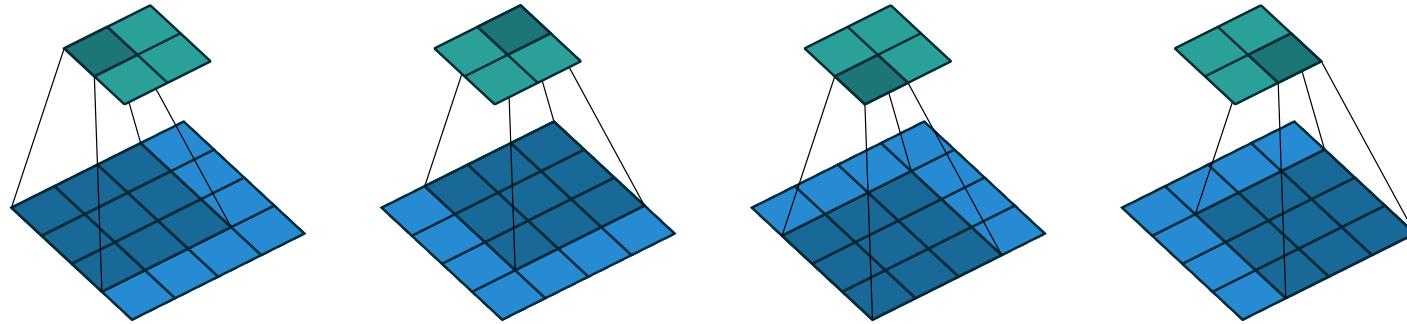


Figure 2.1: (No padding, unit strides) Convolving a 3×3 kernel over a 4×4 input using unit strides (i.e., $i = 4$, $k = 3$, $s = 1$ and $p = 0$).

C=

$$\begin{pmatrix} w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 & 0 & 0 & 0 & 0 \\ 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 \\ 0 & 0 & 0 & 0 & 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} \end{pmatrix}$$

Transposed convolution

Transposed convolution (example)

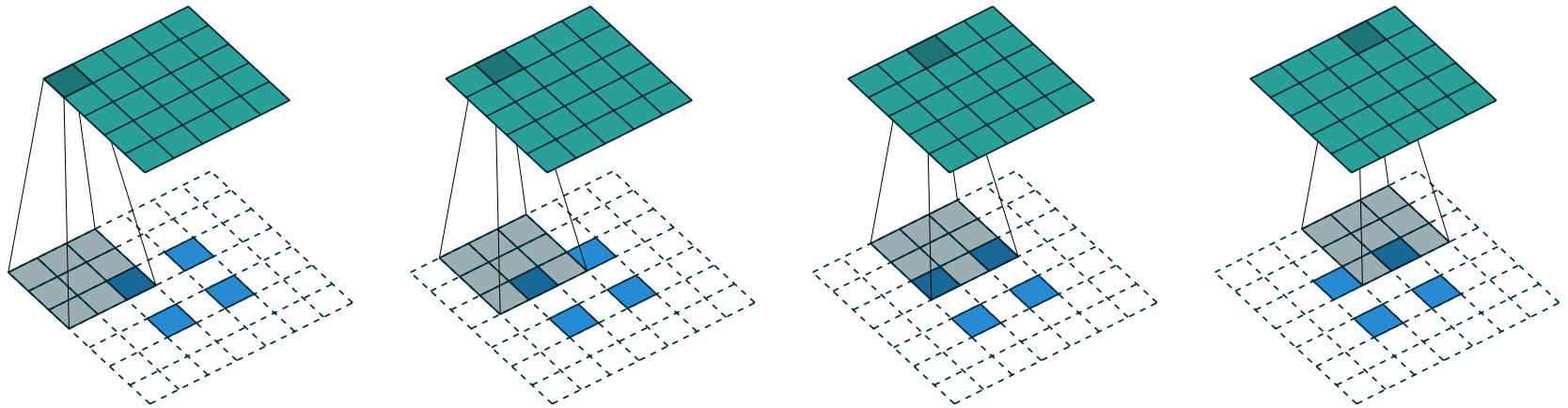
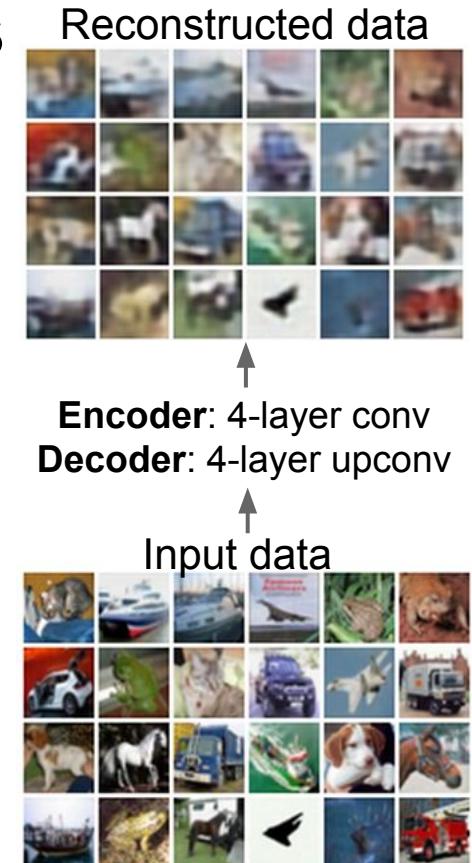
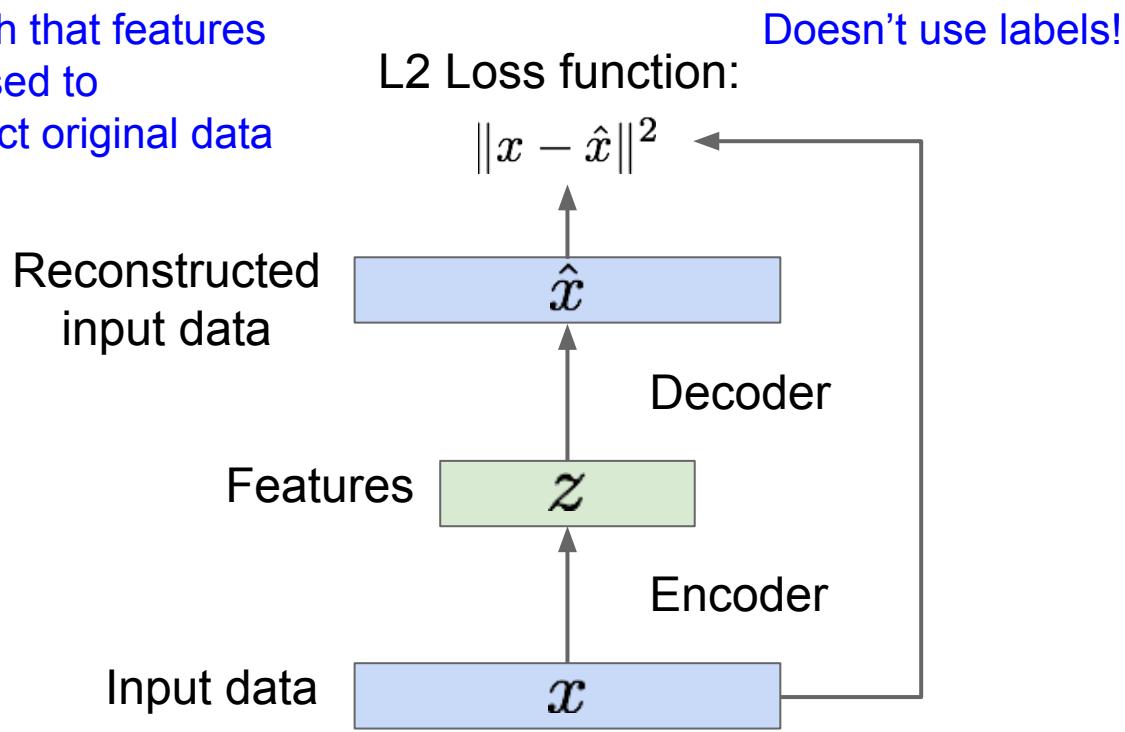


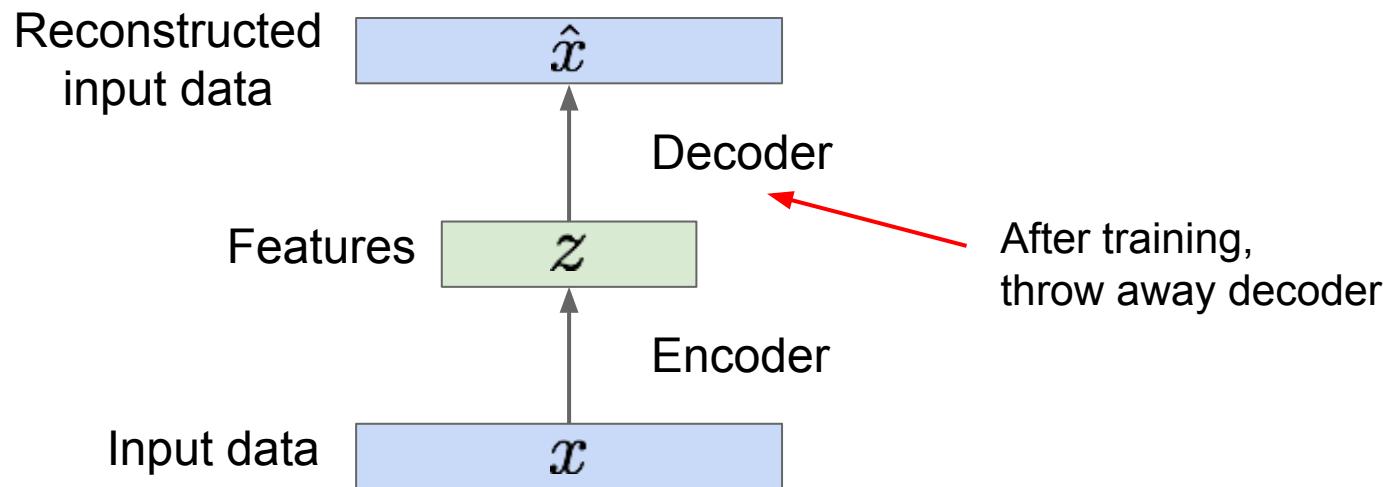
Figure 4.5: The transpose of convolving a 3×3 kernel over a 5×5 input using 2×2 strides (i.e., $i = 5$, $k = 3$, $s = 2$ and $p = 0$). It is equivalent to convolving a 3×3 kernel over a 2×2 input (with 1 zero inserted between inputs) padded with a 2×2 border of zeros using unit strides (i.e., $i' = 2$, $\tilde{i}' = 3$, $k' = k$, $s' = 1$ and $p' = 2$).

Some background first: Autoencoders

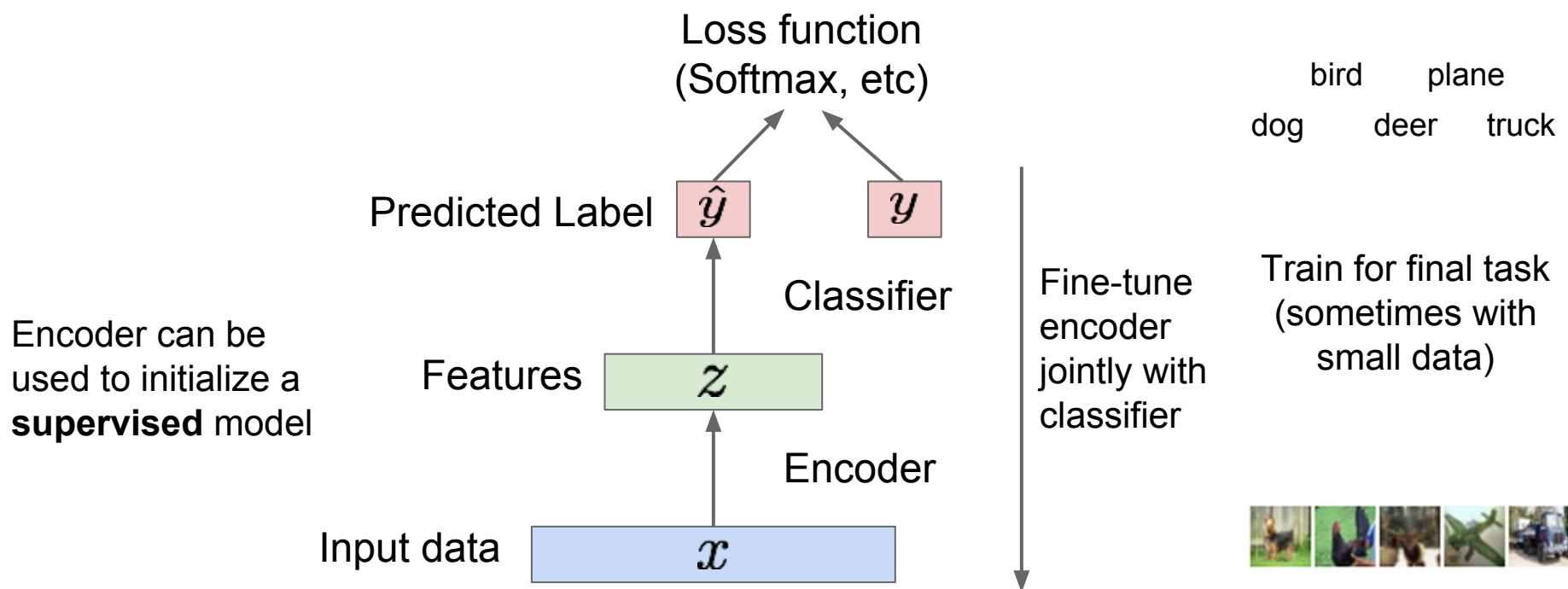
Train such that features
can be used to
reconstruct original data



Some background first: Autoencoders



Some background first: Autoencoders



Variational Autoencoder

Auto-Encoding Variational Bayes

Diederik P. Kingma

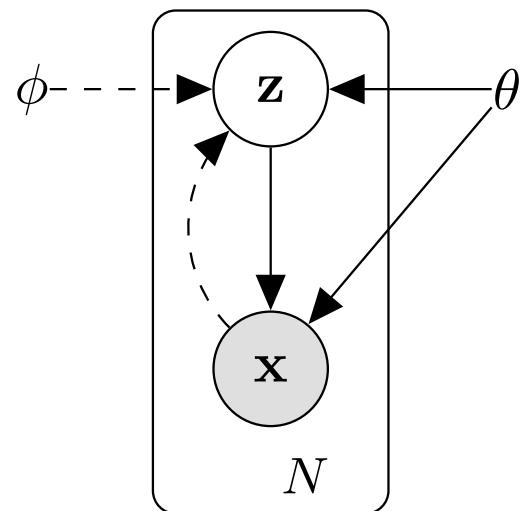
Machine Learning Group
Universiteit van Amsterdam
dpkingma@gmail.com

Max Welling

Machine Learning Group
Universiteit van Amsterdam
welling.max@gmail.com

Graphical model

Probabilistic spin on autoencoders



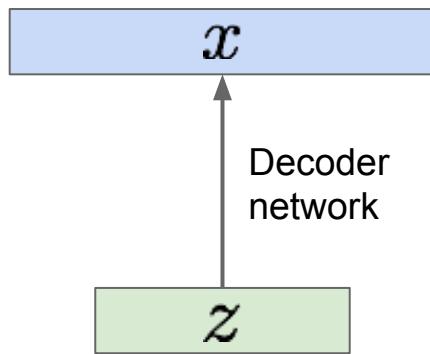
Graphical model

Probabilistic spin on autoencoders

Why we need VAE?

Sample from
true conditional
 $p_{\theta^*}(x | z^{(i)})$

Sample from
true prior
 $p_{\theta^*}(z)$



We want to estimate the true parameters θ^* of this generative model.

How to train the model?

Remember strategy for training generative models from FVBMs. Learn model parameters to maximize likelihood of training data

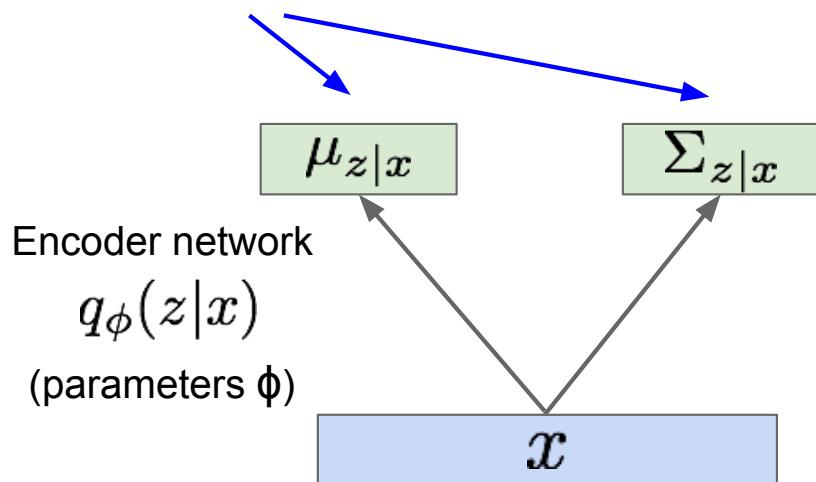
$$p_{\theta}(x) = \int p_{\theta}(z)p_{\theta}(x|z)dz$$

Q: What is the problem with this?
Intractable!

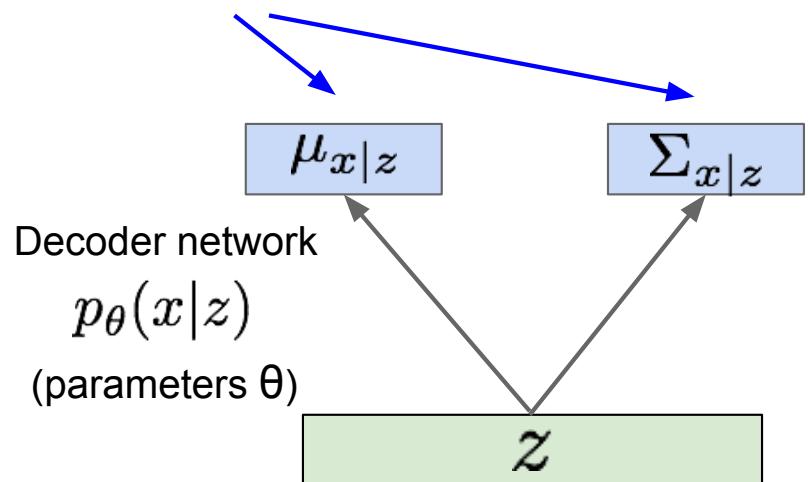
Variational Autoencoder

Since we're modeling probabilistic generation of data, encoder and decoder networks are probabilistic

Mean and (diagonal) covariance of $\mathbf{z} | \mathbf{x}$



Mean and (diagonal) covariance of $\mathbf{x} | \mathbf{z}$



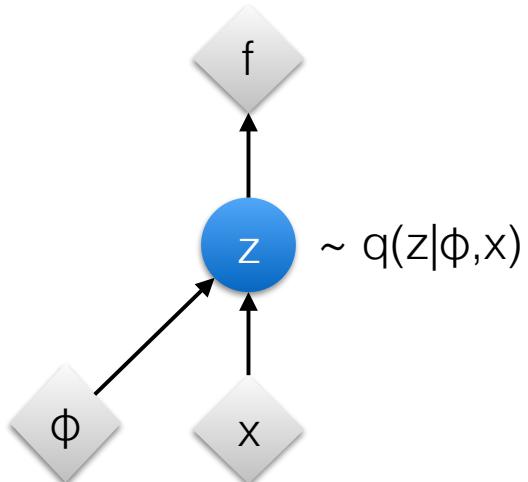
Evidence Lower Bound (ELBO)

ELBO and KL divergence

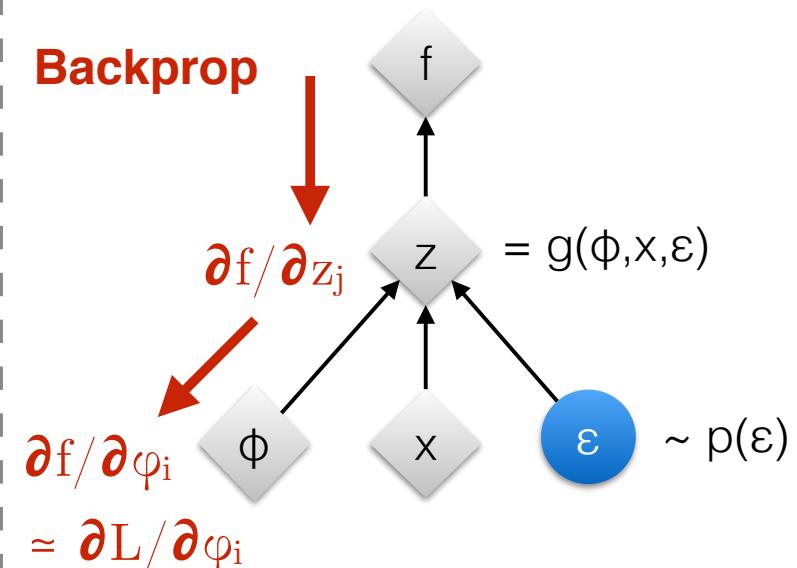
Tractable ELBO

Reparameterization Trick

Original form



Reparameterised form



: Deterministic node



: Random node

[Kingma, 2013]

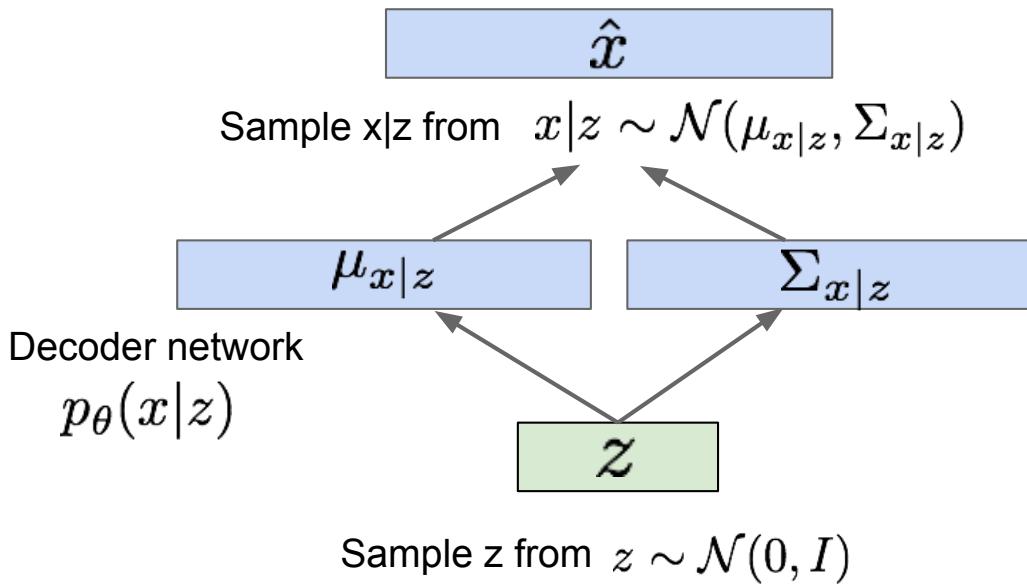
[Bengio, 2013]

[Kingma and Welling 2014]

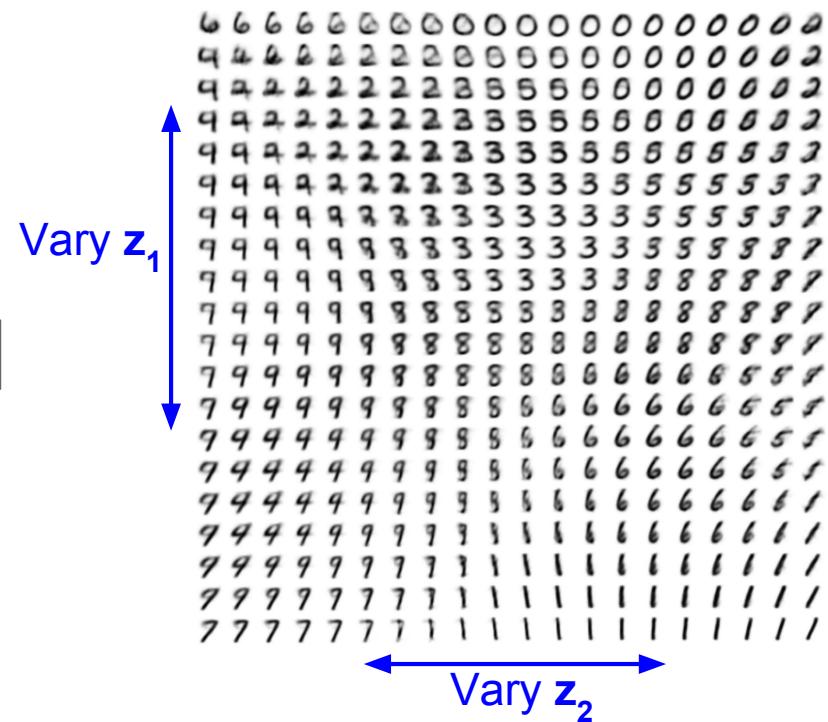
[Rezende et al 2014]

Variational Autoencoders: Generating Data!

Use decoder network. Now sample z from prior!



Data manifold for 2-d z



Kingma and Welling, "Auto-Encoding Variational Bayes", ICLR 2014

Variational Autoencoders: Generating Data!

Diagonal prior on \mathbf{z}
=> independent
latent variables

Different
dimensions of \mathbf{z}
encode
interpretable factors
of variation

Also good feature representation that
can be computed using $q_{\phi}(z|x)$!

Degree of smile

Vary z_1



Vary z_2

Head pose

Kingma and Welling, "Auto-Encoding Variational Bayes", ICLR 2014