Assignment 4

David Felipe Alvear Goyes

3. Variational Autoencoder

KL-divergence
Multivariate gaussian

$$D_{KL}(N(x|\mu_1,\Sigma_1)\,\|\,N(x|\mu_2,\Sigma_2)) = \frac{1}{2}\left[tr(\Sigma_2^{-1}\Sigma_1) + (\mu_2-\mu_1)^T\Sigma_2^{-1}(\mu_2-\mu_1) - D + \log\left[\frac{\det(\Sigma_2)}{\det(\Sigma_1)}\right]\right]$$

→ In variational autoencoders it's common to use the gaussian distribution Parametrized ($\mu=0$, $\Sigma=I$) as the prior for the latent variables. Therefore $\mu_2=0$, $\Sigma_2=I$.

→ $tr(\Sigma_2^{-1}\Sigma_1) = tr(\Sigma_1)$ → Sum of diagonal elements of Σ_1 which is the sum of the variances of each latent dimension

→ $\mu_2=0$ → $(\mu_2-\mu_1)^T\Sigma_2^{-1}(\mu_2-\mu_1)$ → $\mu_1^T\mu_1$ → squared euclidean norm μ_1

→ $\det(\Sigma_2)=1$ → $\log\left(\frac{\det(\Sigma_2)}{\det(\Sigma_1)}\right)$ → $-\log(\det(\Sigma_1))$

Now, with the analyzed changes:

$$D_{KL} = \frac{1}{2}\left[\sum_j^D \sigma_{1j} + \sum_i^D \mu_{1i}^2 - D - \sum_j^D \log(\sigma_{1j})\right]$$

σ_{1j}: diagonal entries Σ_1

μ_{1j}: Mean encoder's output

Pseudo-Code VAE

→ Input x:

→ $\mu, \log var = encode(x)$ → $logvar = \log(\Sigma^2)$

→ $z = Sample\_latent\_space(\mu, logvar)$ → Sample from latent space (Normal dist)

→ $\tilde{x} = decode(z)$ → decode to input space

→ $div\_KL = \frac{1}{2} \sum^{D} exp(logvar) + \mu^2 - 1 - logvar$

→ $reconstruction\_loss = Fn\_loss(x, \tilde{x})$ → Fn_loss → Mean squared error
→ Binary cross Entropy

→ $loss = mean(reconstruction\_loss + div\_KL)$

→ return $\tilde{x}$, loss

4. Recurrent Network

→ Math Notation

$$\begin{cases} h_t = \sigma(W_{ih} x_t + b_{ih} + W_{hh} h_{(t-1)} + b_{hh}) \\ y_t = W_{hy} h_t + b_y \end{cases} \qquad t \in [0, T]$$

$\hat{y} = W_{hy} h_T + b_y$ → for sentiment classification take last timestep to predict

5. CNN for sequence classification

Convolutional Network for sentiment Analysis.

Forward pass.

Let $f: \mathbb{R}^{D \times T} \to \mathbb{R}^{C}$
$$\begin{cases} T: \text{input length} \\ D: \text{Features} \\ C: \text{outputsize} \end{cases}$$

1. Convolution $\to$ $z_i = \sum_d x_i^T \cdot K_{i:i+K} \cdot w_d$ $\to$ K Kernelsize
   w_d Conv filter

   $z \in \mathbb{R}^T$

2. Map output channels $\to$ Compute $z_{ic} = \sum_d x_i^T \cdot K_{i:i+K} \cdot w_{dc}$ $\to$ Map from $TD \to TC$.

3. Max-Pool $\to$ Reduce z using max pooling

   $z_c = \max_i z_{ic}$

4. Softmax $\to$ classification = softmax (z)

Tensor size calculation

* input layer $\to$ Sequence length, channel count equal to features/word
* Initial Conv-layer $\to$ Kernel: 2
  channel: 4
  out_width = $11-2+1=10$
  out_size = 10×4 (width x channels)
* initial Max-Pool layer $\to$ Time based maxpooling $\to$ 4 channels $\to$ 4 outputs
* Second Conv-layer $\to$ Kernel: 4
  channel: 5
  out_width = $11-4+1=8$
  out_size = 8×5 (width x channels)
* Second Max-Pool $\to$ Max Pool per channel, 5 channels

* Concatenation → Combine results from max-pooling layers.
  → 4+5 = 9 (output tensor)

* Fully connected layer → input: 9
  output: 2 → sentiment analysis

* Output layer → size tensor equal to 2 for sentiment analysis.

Calculation of trainable params

Embedding → First Conv → Second Conv → Fully Connected

Embedding	First Conv	Second Conv	Fully Connected
Vocab_size * channels	$f: 2\times2$	$f: 4\times4$	$C = 4+5 = 9$
→ channels = 6	$C = 4$	$C: 5$	output: 2
* What is the size of Vocab?	$P = (2\times6+1)\times4$	$P = (4\times6+1)\times5$	$P = (9\times2)+2$
	$P = 52$	$P = 125$	$P = 20$

↓ Total params: 197