Assignment 4

David Felipe Alvear Goyes

3. Variational Autoencoder

Kl-divergence Multivariate gaussian

$$\text{DKL}(N(x|\mathcal{M}_1, \Sigma_2) || N(x|\mathcal{M}_2, \Sigma_2)) = \frac{1}{2} \left[t_1(\Sigma_2^{-1} \overline{\Sigma}_1) + (\mathcal{M}_2 - \mathcal{M}_1)^T \overline{\Sigma}_2^{-1} (\mathcal{M}_2 - \mathcal{M}_1) - D + \log \left[\frac{\det(\Sigma_2)}{\det(\Sigma_2)} \right] \right]$$

- In variational autoencoders 11's common to use the gaussian distribution Parametrized (M=0, $\Sigma_{=}I$) as the Prior for the latentuariables. Therefore $M_z=0$, $\Sigma_z=I$.
- varphi v
- $\mathcal{L}_{2}^{-0} = (\mathcal{L}_{2} \mathcal{L}_{1})^{T} \Sigma_{2}^{-1} (\mathcal{L}_{2} \mathcal{L}_{1}) \rightarrow \mathcal{L}_{1}^{T} \mathcal{L}_{1} \rightarrow \text{squared evolidean norm } \mathcal{L}_{1}$
- → let(Σ_2)=1 → log(det(Σ_1)) → -log(det(Σ_1))

Now, with the analyzed changes:

$$\mathcal{D}_{KL} = \frac{1}{2} \left[\sum_{j=1}^{p} \delta_{ij} + \sum_{i=1}^{p} \mathcal{U}_{i}^{2} - D - \sum_{j=1}^{p} \lambda_{ij} (\delta_{ij}) \right]$$

 δ_{1j} : diagonal entries Σ_1

M1j: Mean encoder's output

Pseudo-Code VAE

- → Input X:
- → M, Log var = encode (x) -> Loguar= Log([2])
- -> Z = Sample_latent_space(N, loguar) -> Sample from latent space (Normal dist)
- → \hat{X} = decode(z) → decode to input space
- -> $div_{KL} = \frac{1}{2} \stackrel{D}{\geq} exp(logvar) + M^2 1 logvar$
- → leconstruction-loss = Fn-loss (x, x) → Fn-loss → Mean squared error
 → Binary cross Entropy
- loss = mean (reconstruction loss + div_KL)
- return x, loss

4. Recurrent Network

$$\begin{bmatrix} h_{t} = \delta(w_{ih}x_{t} + b_{ih} + w_{hh}h_{(t-1)} + b_{hh}) \\ y_{t} = w_{hy}h_{t} + b_{y} \\ t \in [0,T] \end{bmatrix}$$

ŷ = Wny h_tby → for sentiment classification take last timestep to predict

5. CNN for sequence classification

Convolutional Network for sentiment Analysis.

Forward Pass. Let $f: \mathbb{R}^{D \times T} \to \mathbb{R}^{C}$ $\begin{cases}
T: \text{ input length} \\
D: \text{ Features} \\
C: \text{ output size}
\end{cases}$

1. Convolution → Zi = \(\sum_{d}^{\tau} \tau_{i:i+K} \cdot \wd \rightarrow \tau_{d} \tau_{d Z E DT

2. Map output channels -> Compute Zic = \(\sum_{i} \times \ti_{i:i+k} \times \times \tag{Wdc} \to \tag{Map from } \)

3. Max-Pool -> Reduce Z usign max pooling Zc = max, Zic

4. Softmax -> clasification = Softmax (2)

Tensor Size calculation

* input layer -> Sequence length, channel count equal to features/word

Initial Conv-layer -> Kernel: 2

channel: 4

out_width = 11-2+1=10

out-size = 10x4 (width x channels)

* initial Max-Pool layer -> Time based maxpooling -> 4 channels -> 4 outputs

* Second Convolager -> Kernel: 4

channel: 5

out_width = 11-4+1=8

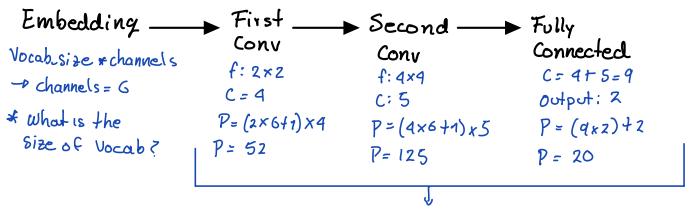
out-size = 8x5 (width x channels)

* Second Max-Pool - Max Pool Per Channel, Schannels

- * Concatenation Combine results from max-pooling layers.
 -> 4+5= 9 (output tensor)
- * Fully connected layer -> Input: q

 output: 2 -> Sentiment analysis
- * Output layer -> Size fensor equal to 2 for sentiment analysis.

Calculation of trainable params



Total params: 197