CSE 676: Assignment #2

Due data: 11:59 PM, April 12, 2019

0.1 Convolutional Neural Networks [3 points]

1) [1 point] Define the convolutional operator for input $\mathbf{x} \in \mathbb{R}^2$ and filter $\mathbf{w} \in \mathbb{R}^2$ as:

$$(\mathbf{x} * \mathbf{w})_{ij} \triangleq \sum_k \sum_\ell \mathbf{x}_{k+i,\ell+j}\, \mathbf{w}_{k\ell} \ .$$

Let $\mathbf{x} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix}$; and $\mathbf{w} = \begin{bmatrix} 4 & 2 \\ 5 & 1 \end{bmatrix}$. 1) Compute the result of $\mathbf{x} * \mathbf{w}$ with padding of size 1 and stride of size 2. 2) What is the relation between input size I, output size O, filter size F, padding size P and stride S?

2) [2 point] For the LeNet architecture, the input are images of size 32×32, the first layer uses a convolution layer with 6 filters, each with a size of 5×5, zero padding and stride of size 1.

   1) What is the output size and how many parameters are there in the first layer?
   2) Propose a way to reduce the number of parameters, and calculate how many parameters are there in your proposed scheme.

0.2 Recurrent Neural Networks [3 points]

- Given the LSTM structure in Figure 1 and the corresponding definition in (1).

$$\begin{pmatrix} \mathbf{i}_t \\ \mathbf{f}_t \\ \mathbf{o}_t \\ \mathbf{g}_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} \left[\begin{pmatrix} \mathbf{W}_1 \\ \mathbf{W}_2 \\ \mathbf{W}_3 \\ \mathbf{W}_4 \end{pmatrix} \begin{pmatrix} \mathbf{h}_{t-1} \\ \mathbf{x}_t \end{pmatrix} \right] \tag{1}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t$$
$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t)$$

Let the loss of an LSTM model be $\mathcal{L}$. Assume we have calculated $\frac{\partial \mathcal{L}}{\partial \mathbf{i}_{t+1}}$, $\frac{\partial \mathcal{L}}{\partial \mathbf{f}_{t+1}}$, $\frac{\partial \mathcal{L}}{\partial \mathbf{o}_{t+1}}$, $\frac{\partial \mathcal{L}}{\partial \mathbf{g}_{t+1}}$, $\frac{\partial \mathcal{L}}{\partial \mathbf{c}_{t+1}}$ and $\frac{\partial \mathcal{L}}{\partial \mathbf{h}_{t+1}}$. Derive gradient formulas for $\frac{\partial \mathcal{L}}{\partial \mathbf{i}_t}$, $\frac{\partial \mathcal{L}}{\partial \mathbf{f}_t}$, $\frac{\partial \mathcal{L}}{\partial \mathbf{o}_t}$, $\frac{\partial \mathcal{L}}{\partial \mathbf{g}_t}$, $\frac{\partial \mathcal{L}}{\partial \mathbf{c}_t}$ and $\frac{\partial \mathcal{L}}{\partial \mathbf{h}_t}$.

0.3 Variational Autoencoder [3 points]

Assume we have the true posterior distribution of latent variable z given data x, $p(z|x)$, as a mixture of two Gaussian, with contour shown as blue curves in Figure 2. In VAE, we

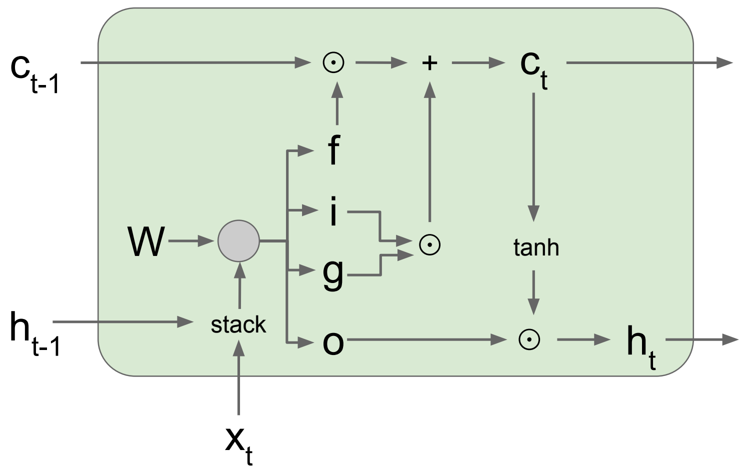

Figure 1: LSTM

use a proposal distribution $q(z|x)$ to approximate $p(z|x)$. The possibly learned distributions $q(z|x)$ are plotted with red curves in Figure 2.

1) [1 points] Point out what form(s) of $q(z|x)$ can be learned with standard VAE. Choose the corresponding plots (a), (b), (c) or (d) in Figure 2. This is potentially a multiple-choice problem.

2) [2 points] Explain your answer.

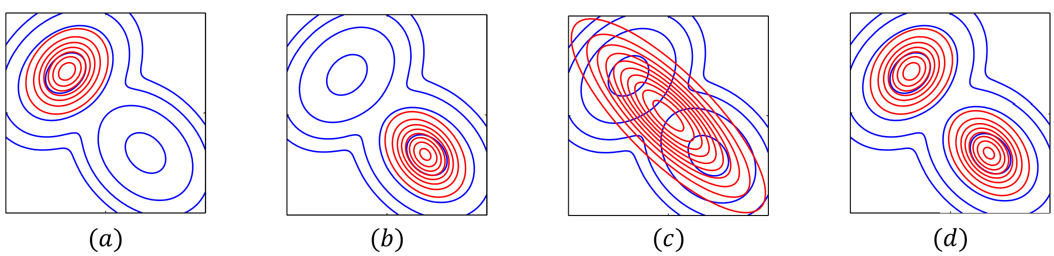

Figure 2: True posterior distribution $p(z|x)$ (blue) and variational distribution $q(z|x)$ (red).

0.4 Generative Adversarial Networks [1 points]

Run any GAN models on the MNIST dataset, and plot out the learning curves of generator and discriminator losses versus the number of epoch. You can reuse any code online.