COMP9444 Neural Networks and Deep Learning

Quiz 1 Answers (Perceptron Learning and Backpropagation)

---

1. What class of functions can be learned by a Perceptron?

   Linearly Separable functions can be learned by a Perceptron.

2. Explain the difference between Perceptron Learning and Backpropagation.

   Perceptron Learning is only used by a Perceptron (one-layer neural network with step activation). Assume the function computed by the Perceptron is $g(w_0 + \Sigma_k w_k x_k)$ where $g()$ is the Heaviside step funtion, and η is the learning rate. If the output is 0 but should have been 1, η is added to the bias w_0, and ηx_k is added to each weight w_k. If the output is 1 but should have been 0, these values are instead subtracted rather than added.

   Backpropagation is a form of gradient descent, which can be applied to multi-layer neural networks provided the activation function is (mostly) differentiable. The derivative $\partial E/\partial w$ of the cost function E with respect to each weight w is calculated, and $\eta\ \partial E/\partial w$ is subtracted from w.

3. When training a Neural Network by Backpropagation, what happens if the Learning Rate is too low? What happens if it is too high?

   If the learning rate is too low, the training will be very slow. If it is too high, the training may become unstable and fail to learn the task successfully.

4. Explain why rescaling of inputs is sometimes necessary for Neural Networks.

   The differential of each weight in the first layer gets multiplied by the value of its corresponding input. Therefore, the network may give undue emphasis to inputs of larger magnitude. Rescaling encourages all inputs to be treated with equal importance.

5. What is the difference between Online Learning, Batch Learning, Mini-Batch Learning and Experience Replay? Which of these methods are referred to as "Stochastic Gradient Descent"?

   For online learning, each training item is presented to the network individually, and the weights are updated using the differentials computed for that item. For batch learning, the differentials for all training items are computed and aggregated, and the weights are updated simultaneously using these aggregated differentials. For mini-batch learning, the differentials for all items in a subset of the training data (called a mini-batch) are computed (perhaps in parallel) and these combined differentials are used to update the weights. For experience replay, items are generated by a separate process (for example, playing a video game) and stored in a database; minibatches are then selected from that database and used to train the network in parallel. The term "Stochastic Gradient Descent" is sometimes used to refer to any method other than pure batch learning, because the order of the training items, or the choice of mini-batch, effectively adds some random noise to the true gradient.

COMP9444 Neural Networks and Deep Learning

Quiz 2 (Probability and Backprop Variations)

This is an optional quiz to test your understanding of the material from Week 2.

---

1. Write the formula for a Gaussian distribution with mean μ and standard deviation σ.

   $P(x) = \mathrm{exp}(-(x-\mu)^2/2\sigma^2) / (\mathrm{sqrt}(2\pi)\sigma)$

2. Write the formula for Bayes' Rule, in terms of a cause A and an effect B.

   $P(A|B) = P(B|A)P(A) / P(B)$

3. Write the formula for the Entropy $H(p)$ of a continuous probability distribution $p()$

   $H(p) = \int_\theta p(\theta)\ (-\log p(\theta))\ d\theta$

4. Write the formula for the Kullback-Leibler Divergence $D_{KL}(p \parallel q)$ between two continuous probability distributions $p()$ and $q()$.

   $D_{KL}(p \parallel q) = \int_\theta p(\theta)\ (\log p(\theta) - \log q(\theta))\ d\theta$

5. Write the formulas for these Loss functions: Squared Error, Cross Entropy, Weight Decay. (remember to define any variables you use)

   Assume z_i is the actual output, t_i is the target output and w_j are the weights.

   Squared Error: $E = ½ \Sigma_i (z_i - t_i)^2$

   Cross Entropy: $E = \Sigma_i (-t_i \log(z_i) - (1- t_i)\log(1- z_i))$

   Weight Decay: $E = ½ \Sigma_j w_j^2$

6. In the context of Supervised Learning, explain the difference between Maximum Likelihood estimation and Bayesian Inference.

   In Maximum Likelihood estimation, the hypothesis $h \in H$ is chosen which maximizes the conditional probability $P(D \mid h)$ of the observed data D, conditioned on h. In Bayesian Inference, the hypothesis $h \in H$ is chosen which maximizes $P(D \mid h)P(h)$, where $P(h)$ is the prior probability of h.

7. Briefly explain the concept of Momentum, as an enhancement for Gradient Descent.

   A running average of the differentials for each weight is maintained and used to update the weights as follows:

   $\delta w = \alpha\delta w - \eta\ dE/dw$
   $w\ = w + \delta w$

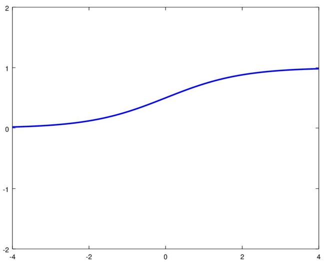
   The constant α with $0 \le \alpha < 1$ is called the momentum.

COMP9444 Neural Networks and Deep Learning

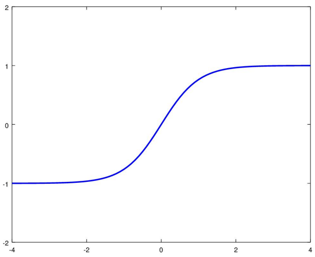
Quiz 3 (Convolutional Networks)

This is an optional quiz to test your understanding of the material from Week 3.

1. Sketch the following activation functions, and write their formula: Sigmoid, Tanh, ReLU.

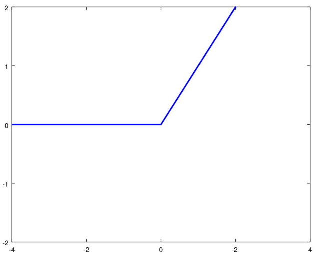


$y = 1/(1 + \exp(-x))$



$y = \tanh(x)$

$= (e^x - e^{-x})/(e^x + e^{-x})$



$y = 0$, if $x \le 0$

$y = x$, if $x > 0$

2. Explain how Dropout is used for neural networks, in both the training and testing phase.

   During each minibatch of training, a fixed percentage (usually, one half) of nodes are chosen to be inactive. In the testing phase, all nodes are active but the activation of each node is multiplied by the same percentage that was used in training.

3. Explain what is meant by Overfitting in neural networks, and list four different methods for avoiding it.

   Overfitting is where the training set error continues to reduce, but the test set error stalls or increases. This can be avoided by

   a. reducing the number of neurons or connections in the network
   b. early stopping, with a validation set
   c. weight decay
   d. dropout

4. Write the formula for the Softmax loss function

   softmax: $E = -(z_i - \log \Sigma_j \exp(z_j))$, where i is the correct class.

5. Write the formula for activation $Z^i_{j,k}$ of the node at location (j,k) in the i^{th} filter of a Convolutional neural network which is connected by weights $K^i_{l,m,n}$ to all nodes in an $M \times N$ window from the L channels in the previous layer, assuming bias weights are included and the activation function is $g()$. How many free parameters would there be in this layer?

   $Z^i_{j,k} = g(b^i + \Sigma_l \Sigma_{0 \le m < M} \Sigma_{0 \le n < N} K^i_{l,m,n} V^l_{j+m,k+n})$

   The number of free parameters is: $F \times (1 + L \times M \times N)$ where F is the number of filters in this layer.

6. If the previous layer has size $J \times K$, and a filter of size $M \times N$ is applied with stride s and zero-padding of width P, what will be the size of the resulting convolutional layer?

   $(1 + (J+2P-M)/s) \times (1 + (K+2P-N)/s)$

7. If max pooling with filter size F and stride s is applied to a layer of size $J \times K$, what will be the size of the resulting (downsampled) layer?

$(1 + (J\text{-}F)/s) \times (1 + (K\text{-}F)/s)$

COMP9444 Neural Networks and Deep Learning

Quiz 4 (Image Processing)

This is an optional quiz to test your understanding of the Image Processing topic from Week 4.

1. List five different Image Processing tasks.

   - image classification
   - object detection
   - object segmentation
   - style transfer
   - generating images
   - generating art
   - image captioning

2. Explain the concept of Data Augmentation, and how it was used in AlexNet.

   Data Augmentation is when additional training items are generated from those originally provided, using domain knowledge. In AlexNet, each original image was randomly cropped in different ways to create images of size 224 × 224. Images can also be reflected left-to-right, and changes can be made to the RGB channels of the images.

3. Explain the problem of vanishing and exploding gradients, and how Weight Initialization can help to prevent it.

   The differentials in a deep neural network tend to grow according to this equation

   $\mathrm{Var}[\partial/\partial x] \approx (\Pi_{i=1}{}^{\mathrm{D}}\ \mathrm{G}_1 n_i{}^{\mathrm{out}}\ \mathrm{Var}[w^{(i)}])\ \mathrm{Var}[\partial/\partial \mathrm{z}]$

   where $w^{(i)}$ are the weights at layer i, $n_i{}^{\mathrm{out}}$ is the number of weights fanning out from each node in layer i, and G_1 estimates the average value of the derivative of the transfer function. If the weights are initialized so that the factor in parentheses corresponding to each layer is approximately 1, then the differentials will remain in a healthy range. Otherwise, they may either grow or vanish exponentially.

4. Describe the Batch Normalization algorithm.

   The mean and variance of the activations $x_k{}^{(i)}$ at layer i over a batch of training items are estimated or pre-computed, and *normalized* activations are calculated for each node
   $\hat{x}_k{}^{(i)} = (x_k{}^{(i)} - \mathrm{Mean}[x_k{}^{(i)}])/\ \mathrm{sqrt}(\mathrm{Var}[x_k{}^{(i)}])$
   These activations are then shifted and rescaled by
   $y_k{}^{(i)} = \beta_k{}^{(i)} + \gamma_k{}^{(i)}\ \hat{x}_k{}^{(i)}$
   where $\beta_k{}^{(i)}$, $\gamma_k{}^{(i)}$ are additional parameters to be learned by backpropagation.

5. Explain the difference between a Residual Network and a Dense Network.

   A Residual Network includes "skip" connections which bypass each pair of consecutive layers. These intermediate layers therefore compute a residual component, which is added to the output from previous layers and corrects their errors, or provides additional details which they were not powerful enough to compute.

   A Dense Network is built from densely connected blocks, separated by convolution and pooling layers. Within a dense block, each layer is connected by shortcut connections to all preceding layers.

6. What is the formula for the $(i, j)^{\text{th}}$ entry G^l_{ij} in the Gram matrix at level l of a convolutional neural network? (remember to define any terms that you use)

   $G^l_{ij} = \Sigma_k F^l_{ik} F^l_{jk}$, where F^l_{ik} is the i^{th} filter at depth l in spatial location k.

7. Explain the difference between Texture Synthesis and Style Transfer (both in their purpose, and their cost function).

   Texture Synthesis aims to produce an image which matches the texture of a given (perhaps, smaller) image. Its cost function is

   $\mathrm{E}_{\text{style}} = (1/4)\ \Sigma_{l=0}^{\mathrm{L}}\ (w_l/N_l^2 M_l^2)\ \Sigma_{i,j}\ (G^l_{ij} - A^l_{ij})^2$

   where:
   w_l is a weighting factor for each layer l
   N_l, M_l are the number of features, and size of the feature maps, in layer l
   G^l_{ij}, A^l_{ij} are the Gram matrices of the original and synthetic image.

   Neural Style Transfer aims to combine the content of one image (x_c) with the style of another, to produce a new image x. Its cost function is

   $\mathrm{E} = \alpha \mathrm{E}_{\text{content}} + \beta \mathrm{E}_{\text{style}}$,
   where $\mathrm{E}_{\text{content}} = (1/2)\ \Sigma_{i,\,k}\ \|F^l_{ik}(x) - F^l_{ik}(x_c)\|^2$

COMP9444 Neural Networks and Deep Learning

Quiz 5 (Recurrent Networks)

This is an optional quiz to test your understanding of Recurrent Networks from Week 5.

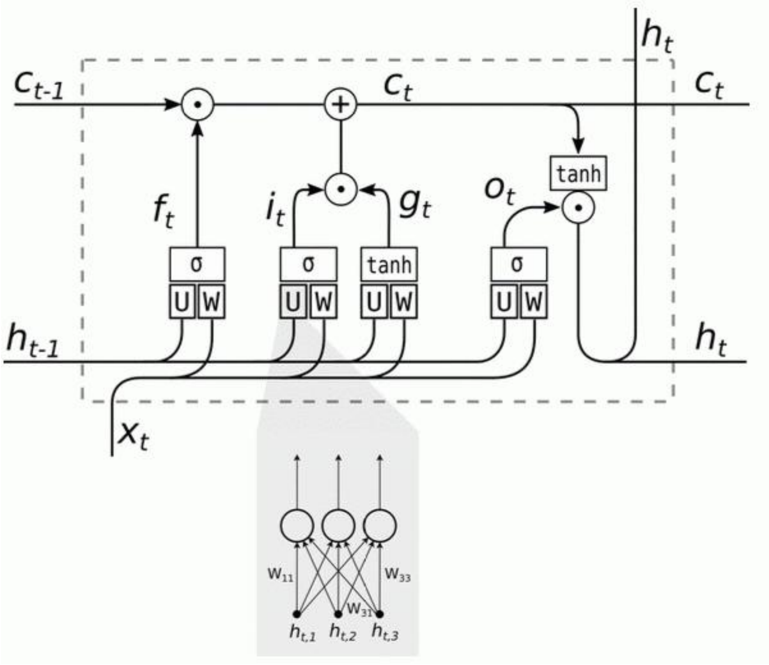
1. Explain the format and method by which input was fed to the NetTalk system, and the target output.

   Characters were fed to NetTalk using a sliding window approach. The characters in a 7-word window were encoded with a 1-hot encoding to form the input of size 7 × 29. The network had 26 outputs - each corresponding to a letter of the phonetic alphabet. The target output was the correct pronunciation of the central character in the input.

2. Explain the role of the context layer in an Elman network.

   The context layer is a copy of the hidden layer at the previous timestep. The hidden layer accepts connections from both the hidden and context layers. This in theory allows the network to retain "state" information for an indefinite period of time.

3. Draw a diagram of an LSTM and write the equations for its operation.

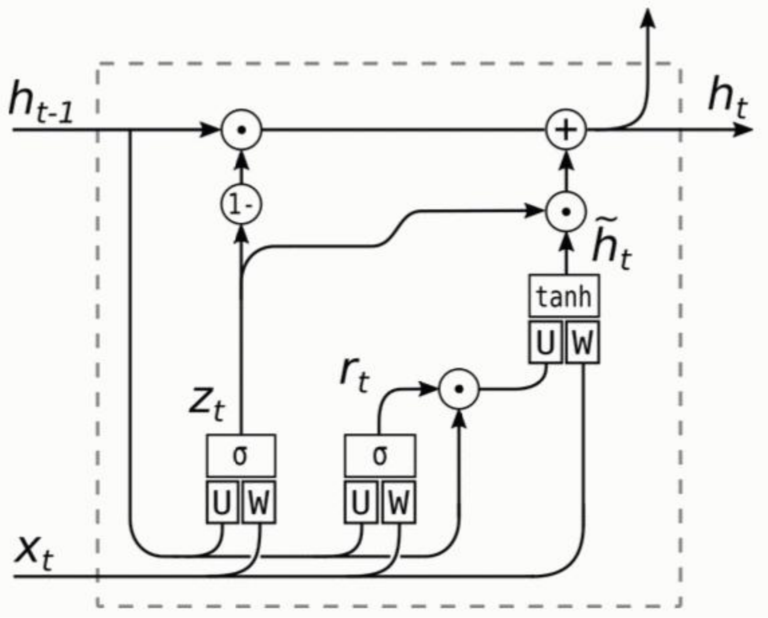


   Gates:
   $f_t = \sigma(W_f\, x_t + U_f\, h_{t-1} + b_f)$ [forget gate]
   $i_t = \sigma(W_i\, x_t + U_i\, h_{t-1} + b_i)$ [input gate]
   $g_t = \tanh(W_g\, x_t + U_g\, h_{t-1} + b_g)$
   $o_t = \sigma(W_o\, x_t + U_o\, h_{t-1} + b_o)$ [output gate]

   State:
   $c_t = c_{t-1} \otimes f_t + i_t \otimes g_t$

   Output:
   $h_t = \tanh(c_t) \otimes o_t$

4. Draw a diagram of a Gated Recurrent Unit and write the equitions for its operation.



Gates:
$z_t = \sigma(W_z\, x_t + U_z\, h_{t-1} + b_z)$
$r_t = \sigma(W_r\, x_t + U_r\, h_{t-1} + b_r)$

Candidate Activation:
$\hat{h}_t = \tanh(W\, x_t + U(r_t \otimes h_{t-1}) + b_h)$

Output:
$h_t = (1 - z_t) \otimes h_{t-1} + z_t \otimes \hat{h}$

5. Briefly describe the problem of *long range dependencies*, and discuss how well each of the following architectures is able to deal with long range dependencies:
   a. sliding window approach
   b. Simple Recurrent (Elman) Network
   c. Long Short Term Memory (LSTM)
   d. Gated Recurrent Unit (GRU)

   For sequence processing tasks, it can happen that the correct output depends on inputs that occurred many timesteps earlier. The sliding window approach is unable to take account of any input beyond the edge of the window. Simple Recurrent Networks can learn medium-range dependencies but may struggle with long range dependencies unless the training data are carefully constructed and the amount of "state" information is limited. LSTMs and GRUs are more successful at learning long range dependencies because they can learn to use some dimensions for short-term processing and others for long-term information.

COMP9444 Neural Networks and Deep Learning

Quiz 6 (Word Vectors)

This is an optional quiz to test your understanding of Word Vectors from Week 5.

1. What are the potential benefits of continuous word representations compared to synonyms or taxonomies?

   Synonyms, antonyms and taxonomy require human effort, may be incomplete, and force discrete choices. Continuous representations have the potential to capture gradations of meaning and more fine-grained relationships between words, as well as being extracted automatically without human involvement.

2. What is meant by the Singular Value Decomposition of a matrix X? What are the special properties of the component matrices? What is the time complexity for computing it?

   The Singular Value Decomposition of X is $X = U\ S\ V^T$ where U, V are unitary (all columns of unit length) and S is diagonal with all entries ≥ 0.
   The time to compute it is proportional to $L \times M^2$ if X is L-by-M and $L \geq M$.

3. What cost function is used to train the word2vec skip-gram model? (remember to define any symbols you use)

   If the text is $w_1 \ldots w_T$ then the cost function is

   $-(1/T) \sum_{t=1}^{T} \sum_{-c \leq r \leq c,\, r \neq 0} \log \mathrm{prob}(w_{t+r} \mid w_t)$

4. Explain why full softmax may not be computationally feasible for word-based language processing tasks.

   The number of outputs is equal to the total number of words in the lexicon (approximately 60,000) and all of them would need to be evaluated at every step.

5. Write the formula for Hierarchical Softmax and explain the meaning of all the symbols.

   $\mathrm{prob}(w = w_t) = \prod_{j=1}^{L(w)-1} \sigma(\mathbf{[}n(w, j+1) = \mathrm{child}(n(w, j))\mathbf{]}\ \mathbf{v'}_{n(w,\, j)}{}^{\mathrm{T}}\ \mathbf{h})$

   $n(w,1), \ldots, n(w,L(w))$ are the nodes along the path in a Binary Search Tree from the root to w_t
   h = hidden unit activations, $\sigma(u) = 1/(1 + \exp(-u))$,
   $\mathbf{[}n' = \mathrm{child}(n)\mathbf{]} = +1$, if n' is left child of node n; -1, otherwise.

6. Write the formula for Negative Sampling and explain the meaning of all the symbols.

   $E = -\log \sigma(\mathbf{v'}_{j*}{}^{\mathrm{T}}\ \mathbf{h}) - \Sigma_{j \in Wneg} \log \sigma(-\mathbf{v'}_j{}^{\mathrm{T}}\ \mathbf{h})$
   j* = target word, *W*neg = set of negative examples drawn from some distribution

7. From what probability distribution are the negative examples normally drawn?

   $P(w) = U(w)^{3/4}/Z$,
   $U(w)$ = Unigram distribution determined by previous word,
   Z = normalizing constant.

COMP9444 Neural Networks and Deep Learning

Quiz 7 (Reinforcement Learning)

This is an optional quiz to test your understanding of the Reinforcement Learning material from Week 7.

---

1. Explain the difference between the following paradigms, in terms of what is presented to the agent, and what the agent aims to do:
   - Supervised Learning
   - Unsupervised Learning
   - Reinforcment Learning

     - Supervised Learning: Each training item includes an input and a target output. The aim is to predict the output, given the input (for the training set as well as an unseen test set).
     - Unsupervised Learning: Each training item consists of only an input (no target value). The aim is to learn hidden features, or to infer whatever structure you can, from the data (input items).
     - Reinforcement Learning: An agent chooses actions in a simulated environment, observing its state and receiving rewards along the way. The aim is to maximize the cumulative reward.

2. Describe the elements (sets and functions) that are needed to give a formal description of a reinforcement learning environment. What is the difference between a deterministic environment and a stochastic environment?

   Formally, a reinforcement learning environment is defined by a set S of states, a set A of actions, a transition function δ and a reward function R.
   For a deterministic environment, δ and R are single-valued functions:
   $\delta : S \times A \rightarrow S$ and $R : S \times A \rightarrow \mathbf{R}$
   For a stochastic environment, δ and/or R are not single-valued, but instead define a probability distribution on S or $\mathbf{R}$.

3. Name three different models of optimality in reinforcement learning, and give a formula for calculating each one.

Finite horizon reward:	$\Sigma_{0 \le i < h}\, r_{t+i}$
Infinite discounted reward:	$\Sigma_{i \ge 0}\, \gamma^i\, r_{t+i},\quad 0 \le \gamma < 1$
Average reward:	$\lim_{h \rightarrow \infty} (1/h)\, \Sigma_{0 \le i < h}\, r_{t+i}$

4. What is the definition of:
   a. the optimal policy
   b. the value function
   c. the Q-function?

   a. The optimal policy is the function $\pi^*: S \rightarrow A$, which maximizes the infinite discounted reward.
   b. The value function $V^{\pi}(s)$ is the expected infinite discounted reward obtained by following policy π starting from state s. If $\pi = \pi^*$ is optimal, then $V^*(s) = V^{\pi^*}(s)$ is the maximum (expected) infinite discounted reward obtainable from state s.
   c. The Q-function $Q^{\pi}(s,a)$ is the expected infinite discounted reward received by an agent who begins in state s, first performs action a and then follows policy π for all subsequent timesteps. If $\pi = \pi^*$ is optimal, then $Q^*(s,a) = Q^{\pi^*}(s,a)$ is the maximum

(expected) discounted reward obtainable from s, if the agent is forced to take action a in the first timestep but can act optimally thereafter.

5. Assuming a stochastic environment, discount factor γ and learning rate of η, write the equation for
   a. Temporal Difference learning TD(0)

   $$V(s_t) \leftarrow V(s_t) + \eta\ [r_t + \gamma V(s_{t+1}) - V(s_t)]$$

   b. Q-Learning

   $$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \eta\ [r_t + \gamma\ \max_b Q(s_{t+1}, b) - Q(s_t, a_t)]$$

   Remember to define any symbols you use.

   s_t = state at time t, a_t = action performed at time t,
   r_t = reward received at time t, s_{t+1} = state at time $t+1$.

COMP9444 Neural Networks and Deep Learning

Quiz 8 Deep RL and Unsupervised Learning

This is an optional quiz to test your understanding of Deep RL and Unsupervised Learning.

1. Write out the steps in the REINFORCE algorithm, making sure to define any symbols you use.

```
for each trial
    run trial and collect states $s_t$, acions $a_t$ and reward $r_{\text{total}}$
    for $t$ = 1 to length(trial)
        $\theta \leftarrow \theta + \eta(r_{\text{total}} - b)\, \nabla_\theta \log \pi_\theta(a_t \mid s_t)$
    end
end
```

θ = parameters of policy, η = learning rate,
r_{total} = total reward received during trial,
b = baseline (constant), ∇_θ = gradient with respect to θ,
$\pi_\theta(a \mid s)$ = probability of performing action a in state s.

2. In the context of Deep Q-Learning, explain the following:

   a. Experience Replay

   The agent(s) choose actions according to their current Q-function, using an ε-greedy strategy, and contribute to a central database of experiences in the form (s_t, a_t, r_t, s_{t+1}). Another thread samples experiences asynchronously from the experience database, and updates the Q-function by gradient descent, to minimize

   $[r_t + \gamma \max_b Q_w(s_{t+1},b) - Q_w(s_t,a_t)]^2$

   b. Double Q-Learning

   Two sets of Q values are maintained. The current Q-network w is used to select actions, and a slightly older Q-network $\bar{w}$ is used for the target value.

3. What is the Energy function for these architectures:

   a. Boltzmann Machine
   b. Restricted Boltzmann Machine

   Remember to define any variables you use.

   a. Boltzmann Machine

   $\mathrm{E}(x) = -(\Sigma_{i<j}\, x_i\, w_{ij}\, x_j + \Sigma_i\, b_i\, x_i)$

   where x_i = activation of node i (0 or 1)

   b. Restricted Boltzmann Machine

   $\mathrm{E}(v, h) = -(\Sigma_i\, b_i\, v_i + \Sigma_j\, c_j\, h_j + \Sigma_{i,j}\, v_i\, w_{ij}\, h_j)$

   where v_i = visible unit activations, h_j = hidden unit activations

4. The Variational Auto-Encoder is trained to maximize

$\mathbf{E}_{z \sim q_\phi(z \mid x^{(i)})} [\log p_\theta(x^{(i)} \mid z)] \; - \; \mathrm{D}_{\mathrm{KL}}(q_\phi(z \mid x^{(i)}) \,\|\, p(z))$

Briefly state what each of these two terms aims to achieve.

The first term enforces that any sample z drawn from the conditional distribution $q_\phi(z \mid x^{(i)})$ should, when fed to the decoder, produce something approximationg $x^{(i)}$.
The second term encourages the distribution $q_\phi(z \mid x^{(i)})$ to approximate the Normal distribution $p(z)$ (by minimizing the KL-divergence between the two distributions)

5. Generative Adversarial Networks traditionally made use of a two-player zero-sum game between a Generator G_θ and a Discriminator D_ψ, to compute

   $\min_\theta \max_\psi (V(G_\theta, D_\psi))$

   a. Give the formula for $V(G_\theta, D_\psi)$.

      $V(G_\theta, D_\psi) = \mathbf{E}_{x \sim p_{\mathrm{data}}} [\log \mathrm{D}_\psi(x)] + \mathbf{E}_{z \sim p_{\mathrm{model}}} [\log(1 - D_\psi(G_\theta(z)))]$

   b. Explain why it may be advantageous to change the GAN algorithm so that the game is no longer zero-sum, and write the formula that the Generator would try to maximize in that case.

      The quality of the generated images tends to improve if the Generator instead tries to maximize

      $\mathbf{E}_{z \sim p_{\mathrm{model}}} [\log(D_\psi(G_\theta(z)))]$

      This forces the Generator to put emphasis on improving the poor-quality images, rather than taking the images that are already good and making them slightly better.

6. In the context of GANs, briefly explain what is meant by *mode collapse*, and list three different methods for avoiding it.

   Mode collapse is when the Generator produces only a small subset of the desired range of images, or converges to a single image (with minor variations). Methods for avoiding mode collapse include: Conditioning Augmentation, Minibatch Features and Unrolled GANs.