For all questions, show your work!

1. **(4 points) Backpropagation)**
   In this question we focus on the guarantees of learning with error backpropagation. Consider training a network to read images (2828 pixels) of handwritten digits (i.e. MNIST). The network consists of 28 28 = 784 input units, a hidden layer of logistic units, and a softmax group of 10 units as the output layer. The objective function is the cross-entropy loss. We have many training cases and we always compute our weight update based on the entire training set, using the error backpropagation algorithm. We use a learning rate thats small enough for all practical purposes, but not so small that the network doesnt learn. We stop when the weight update becomes zero. For each of the following questions, answer yes or no, and explain very briefly (one short sentence can be enough).

   (a) Does this training set-up guarantee that the weight vector gets closer, on every step, to the weight vector that gives the smallest possible loss value (or to one such weight vector if there are multiple)?

   (b) Does this training set-up guarantee that the loss gets better (i.e. smaller) on every step?

   (c) Does this training set-up guarantee that eventually well reach a weight vector that gives the smallest possible loss value?

   (d) We use our network to classify images by simply seeing which of the 10 output units gets the largest probability when the network is presented with the image, and declaring the number of that output unit to be the networks guessed label. Does our training set-up guarantee that the number of mistakes that the network makes on training data (by following this classification strategy) never increases?

2. **(6 points) Invariant and Equivariant Representations**

   (a) What is the difference between an invariant representation and an equivariant representation? Explain in 2-3 sentences.

   (b) Can you imagine a representation that is both invariant and equivariant? If so, describe an example of such a feature.

   (c) Is the maxout unit best described as a invariant or an equivariant feature? Explain your answer.

3. **(12 points) Optimization and Regularization**

   (a) Compare and contrast (providing specific differences, advantages and disadvantages) the following optimization algorithms: Adagrad, RMSprop, Adadelta.

   (b) What are the differences between standard Momentum and Nesterov Momentum?

   (c) Explain why early-stopping prevents overfitting? Would you expect it to work in the context of linear regression?

4. **(12 points) Restricted Boltzmann Machine and Deep Boltzmann Machine**
Consider an undirected probabilistic model over three binary random vectors: $\boldsymbol{x} \in \{0,1\}^l$,$\boldsymbol{y} \in \{0,1\}^m$,$\boldsymbol{z} \in \{0,1\}^n$. As an undirected graphical model, the joint probability distribution is given by $P(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = \frac{1}{Z} \exp\left(-E(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})\right)$ following energy function:

$$E(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = -\boldsymbol{x}^\top W \boldsymbol{y} - \boldsymbol{x}^\top U \boldsymbol{z} - \boldsymbol{y}^\top V \boldsymbol{z} - a\boldsymbol{x}^\top \boldsymbol{x} - b\boldsymbol{y}^\top \boldsymbol{y} - c\boldsymbol{z}^\top \boldsymbol{z} \tag{1}$$

where a, b, c are each scalar offsets and W, U, Z are weight matrices of the appropriate size (e.g. W is an $l \times m$ matrix).

   (a) Derive the conditional distributions $P(\boldsymbol{x} \mid \boldsymbol{y}, \boldsymbol{z})$, $P(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{z})$ and $P(\boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{y})$.

   (b) Describe an efficient method of sampling from the joint probability distribution. Be specific in your description and indicate why your algorithm is efficient.

   (c) Let us consider $\boldsymbol{x}$ as the visible or observed variable. Is the posterior distribution(i.e. $P(\boldsymbol{z}, \boldsymbol{y} \mid \boldsymbol{x})$) tractable to compute, like the RBM, or is it intractable, like the DBM? Explain your answer.

   (d) Is this model amenable to learning via the Contrastive Divergence (CD) algorithm? If so, provide a description of how one could apply CD to this model, if not, explain why. In either case, a few sentences should suffice.

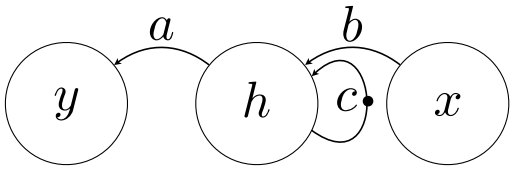
5. **(4 points) Variational Autoencoder (VAE)**

   (a) The Variational Autoencoder represents the standard variational lower bound as the combination of a reconstruction term and a regularization term. Starting with the variational lower bound below, derive these two terms, specifying which is the reconstruction term and which is the regularization term.

$$\mathcal{L}(q) = \int q(\boldsymbol{z} \mid \boldsymbol{x}) \log \left(\frac{p(\boldsymbol{x}, \boldsymbol{z})}{q(\boldsymbol{z} \mid \boldsymbol{x})} \right) \, d\boldsymbol{z} \tag{2}$$

   (b) Both the Variational Autoencoder and the Deep Boltzmann Machine use variational approximate inference. In class, we discussed that one was a parametric application of variational inference and the other uses a more standard non-parametric application of variational inference. Explain what was meant by this and specify which method (VAE or DBM) corresponds to what kind of variational inference (parametric or non-parametric).

6. **(12 points) Backpropagation Through Time**
Consider the Recurrent Neural Network shown below. The arrows with the dots on them indicate delays by one time step (i.e. connection associated with weights b and c). Arrows without dots indicate no time delay (i.e. connection associated with weight a). Units h_0 and h_1 both have a logistic sigmoid activation function.



(a) Plot the computational graph for this RNN, i.e. unrolled the RNN in time for 3 time steps **(from $t = 0$ to $t = 2$)**. Indicate hidden unit activations with superscripts that denote time (e.g.$h_0^{(t=0)}, h_0^{(t=1)}, \dots$) and label all connections with their corresponding weight labels.

(b) Let's consider training this model using stochastic gradient descent. Let's also consider a training data point consisting of input/output pairs $(\boldsymbol{x}, y)$ where $\boldsymbol{x} \in [0,1]^2$ and $y \in [0,1]$. Consider setting $h_0^{(t=0)} = x_0$ and $h_0^{(t=1)} = x_1$ and consider the output of the model to be $h_0^{(t=2)}$, that is our objective function is $C = \frac{1}{2}(h_0^{(t=2)} - y)^2$. Compute the gradient necessary to update the parameter a given this data point, i.e. Compute $\frac{\partial C}{\partial w_1}$.