ELL409: Machine Intelligence and Learning

Major Test, Form: A

Maximum marks: 24

(Answer all questions on this question paper. Use the answer script only for working; it will not be graded. Read all section-specific instructions carefully.)

Name: _____

Entry Number: _____

Section 1. Multiple choice questions

Instructions: Each question may have any number of correct choices. Clearly mark (tick) all choices you believe to be correct (1 mark for each correct choice, -0.5 for each incorrect choice).

1. Which of the following are reasonable ways to select the number of clusters K for K-means?
   (a) Minimising the distortion function J as a function of K
   (b) Set a threshold τ, pick the smallest K such that $J < \tau$
   (c) Set a threshold τ, pick the smallest K such that the reduction in J from $K-1$ to K clusters is less than τ
   (d) Set a threshold τ, pick the smallest K such that the reduction in J from K to $K+1$ clusters is less than τ
2. Which of the following might be valid reasons for preferring a neural network over an SVM?
   (a) An neural net effectively applies a non-linear transformation on the input space; an SVM cannot.
   (b) The model size (number of parameters) for a neural net is fixed in advance, whereas for an SVM it depends on the number of support vectors and hence on the training data.
   (c) A neural net should not get stuck in local minima, unlike an SVM.
   (d) Neural nets extend more naturally to multi-class classification than SVMs.
   (e) Neural nets have an inbuilt regularisation mechanism to avoid overfitting, unlike SVMs.

Section 2. Numerical/Short-answer questions

Instructions: Please write *only the final answers* on this question paper, as concisely as possible, in the space provided for each item. The provided answer script should be used for all working (where applicable), but will not be graded. However, in case of any doubt regarding your answers, we may refer to the answer script to check your working. So please try to write out your working as clearly as possible.

3. Suppose you have run K-means on a labeled data set (without using the labels during clustering), containing 4 classes with 300 images from *each* class. Now, you seek to assess the accuracy of your clustering by assigning to all points in a given cluster a 'predicted label', which is taken to be the true label found most frequently within that cluster. You then report as the accuracy the overall fraction of points which are assigned to the correct label as per this labeling.

   (a) Assuming you have taken the number of clusters K to be 20 or fewer, what is the lowest possible accuracy this could give you? **[1]**

   (b) If we remove this upper bound on K, are there values of K for which this lowest possible accuracy will be greater than the above? If so, what is the smallest value of K for which this will happen, for the given data set? **[1]**

   (c) What is the smallest K at which 100% accuracy can theoretically be achieved for the given data set? **[0.5]**

(d) Why is picking K so as to maximise this notion of accuracy not a good way of determining the number of clusters? **[0.5]**

4. The idea behind gradient descent is that on each iteration, you shift the model parameters by a small amount in a direction (that of the negative gradient) so as to decrease the error. Hence, in theory the error should get lower the longer you run gradient descent for. So why is early stopping often used in training neural networks? Explain as precisely as you can. **[2]**

5. Which supervised learning or prediction technique would you use if you knew the underlying joint distribution of your features and labels, *i.e.*, $p(\mathbf{x}, t)$? Give the precise mathematical model $y(\mathbf{x})$ for it. **[2]**

6. We have seen that the hidden units in neural nets can implement any suitable non-linear activation function. One of the most widely used activation functions nowadays is the *Leaky ReLU* (Rectified Linear Unit), which we will define as follows:

$$LR(a) = \begin{cases} a & \text{if } a > 0 \\ 0.01a & \text{otherwise} \end{cases}$$

Consider a single hidden unit called z, which implements the above activation function. Take z to be connected to D input features $x_1, x_2, ..., x_D$, with corresponding weights $w_1, w_2, ..., w_D$.

(a) Draw a picture of the unit z with all its incoming connections and the corresponding input units. Also include the bias term w_0 in your picture via a dummy input variable. All units and weights should be labeled clearly. You picture should include the expression for the value of z as a function of the input units. **[1.5]**

(b) For backpropagation, we wish to calculate the error gradient with respect to each of these weights just defined, *i.e.*, $\frac{\partial E(\mathbf{w})}{\partial w_i}, i \in \{0, 1, 2, ...D\}$. Apply the chain rule using z as the intermediate variable, to write this partial derivative as a product of two partial derivatives. Where will the value of the first one come from? **[1]**

(c) Here we wish to show the calculation for the second partial derivative obtained via the above chain rule. To do this, we will first need to obtain the derivative of the *Leaky ReLU* function with respect to its argument. Write down (in an appropriate form) the value of this derivative, *i.e.*, $\frac{\partial LR(a)}{\partial a}$. Is there any value of a where it is undefined? **[2]**

(d) Now write down (again, in appropriate form) the entire expression for the value of the second partial derivative from the above chain rule. **[1.5]**

(e) Is there any specific situation in which backpropagation through a Leaky ReLU hidden unit would be theoretically ill-defined? If so, what is it, and why do you think Leaky ReLUs are still usable in practice? **[1.5]**

7. Suppose you run PCA on a high-dimensional labeled data set, and find that only a small number of principal components capture most of the variance (say $> 90\%$). Is it always safe to replace the original features with the smaller number of principal components, for the purposes of supervised learning? If yes, give a general justification. If no, explain using a simple counter-example. **[2]**

8. Consider a GMM with K mixture components being fitted via the EM algorithm for likelihood maximisation, to a given data set $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$. Suppose, at some point during the execution of the EM algorithm, we get for the k^{th} component that $\boldsymbol{\mu}_k \to \mathbf{x}_n$ (for some $n \in \{1, ..., N\}$) and $\Sigma_k \to \mathbf{0}$.

(a) In this case, what would you expect the corresponding value of π_k to be? Why? **[1]**

(b) What will be the corresponding value of the likelihood $p(\mathbf{X})$? Mention the reason why. **[1]**

(c) Is this a solution that the EM algorithm could converge towards? Why or why not? **[0.5]**

(d) Would this be a desirable solution to obtain? Why or why not? If not, can you suggest some way of avoiding it? **[2]**

ELL409: Machine Intelligence and Learning

Major Test, Form: B

Maximum marks: 24

(Answer all questions on this question paper. Use the answer script only for working; it will not be graded. Read all section-specific instructions carefully.)

Name: _____

Entry Number: _____

Section 1. Multiple choice questions

Instructions: Each question may have any number of correct choices. Clearly mark (tick) all choices you believe to be correct (1 mark for each correct choice, -0.5 for each incorrect choice).

1. Which of the following are reasonable ways to select the number of clusters K for K-means?
   (a) Minimising the distortion function J as a function of K
   (b) Set a threshold τ, pick the smallest K such that $J < \tau$
   (c) Set a threshold τ, pick the smallest K such that the reduction in J from $K-1$ to K clusters is less than τ
   (d) Set a threshold τ, pick the smallest K such that the reduction in J from K to $K+1$ clusters is less than τ
2. Which of the following might be valid reasons for preferring a neural network over an SVM?
   (a) An neural net effectively applies a non-linear transformation on the input space; an SVM cannot.
   (b) The model size (number of parameters) for a neural net is fixed in advance, whereas for an SVM it depends on the number of support vectors and hence on the training data.
   (c) A neural net should not get stuck in local minima, unlike an SVM.
   (d) Neural nets extend more naturally to multi-class classification than SVMs.
   (e) Neural nets have an inbuilt regularisation mechanism to avoid overfitting, unlike SVMs.

Section 2. Numerical/Short-answer questions

Instructions: Please write *only the final answers* on this question paper, as concisely as possible, in the space provided for each item. The provided answer script should be used for all working (where applicable), but will not be graded. However, in case of any doubt regarding your answers, we may refer to the answer script to check your working. So please try to write out your working as clearly as possible.

3. Suppose you have run K-means on a labeled data set (without using the labels during clustering), containing 5 classes with 200 images from *each* class. Now, you seek to assess the accuracy of your clustering by assigning to all points in a given cluster a 'predicted label', which is taken to be the true label found most frequently within that cluster. You then report as the accuracy the overall fraction of points which are assigned to the correct label as per this labeling.

   (a) Assuming you have taken the number of clusters K to be 20 or fewer, what is the lowest possible accuracy this could give you? **[1]**

   (b) If we remove this upper bound on K, are there values of K for which this lowest possible accuracy will be greater than the above? If so, what is the smallest value of K for which this will happen, for the given data set? **[1]**

   (c) What is the smallest K at which 100% accuracy can theoretically be achieved for the given data set? **[0.5]**

(d) Why is picking K so as to maximise this notion of accuracy not a good way of determining the number of clusters? **[0.5]**

4. The idea behind gradient descent is that on each iteration, you shift the model parameters by a small amount in a direction (that of the negative gradient) so as to decrease the error. Hence, in theory the error should get lower the longer you run gradient descent for. So why is early stopping often used in training neural networks? Explain as precisely as you can. **[2]**

5. Which supervised learning or prediction technique would you use if you knew the underlying joint distribution of your features and labels, *i.e.*, $p(\mathbf{x}, t)$? Give the precise mathematical model $y(\mathbf{x})$ for it. **[2]**

6. We have seen that the hidden units in neural nets can implement any suitable non-linear activation function. One of the most widely used activation functions nowadays is the *Leaky ReLU* (Rectified Linear Unit), which we will define as follows:

$$LR(a) = \begin{cases} a & \text{if } a > 0 \\ 0.01a & \text{otherwise} \end{cases}$$

Consider a single hidden unit called z, which implements the above activation function. Take z to be connected to D input features $x_1, x_2, ..., x_D$, with corresponding weights $w_1, w_2, ..., w_D$.

(a) Draw a picture of the unit z with all its incoming connections and the corresponding input units. Also include the bias term w_0 in your picture via a dummy input variable. All units and weights should be labeled clearly. You picture should include the expression for the value of z as a function of the input units. **[1.5]**

(b) For backpropagation, we wish to calculate the error gradient with respect to each of these weights just defined, *i.e.*, $\frac{\partial E(\mathbf{w})}{\partial w_i}, i \in \{0, 1, 2, ...D\}$. Apply the chain rule using z as the intermediate variable, to write this partial derivative as a product of two partial derivatives. Where will the value of the first one come from? **[1]**

(c) Here we wish to show the calculation for the second partial derivative obtained via the above chain rule. To do this, we will first need to obtain the derivative of the *Leaky ReLU* function with respect to its argument. Write down (in an appropriate form) the value of this derivative, *i.e.*, $\frac{\partial LR(a)}{\partial a}$. Is there any value of a where it is undefined? **[2]**

(d) Now write down (again, in appropriate form) the entire expression for the value of the second partial derivative from the above chain rule. **[1.5]**

(e) Is there any specific situation in which backpropagation through a Leaky ReLU hidden unit would be theoretically ill-defined? If so, what is it, and why do you think Leaky ReLUs are still usable in practice? **[1.5]**

7. Suppose you run PCA on a high-dimensional labeled data set, and find that only a small number of principal components capture most of the variance (say $> 90\%$). Is it always safe to replace the original features with the smaller number of principal components, for the purposes of supervised learning? If yes, give a general justification. If no, explain using a simple counter-example. **[2]**

8. Consider a GMM with K mixture components being fitted via the EM algorithm for likelihood maximisation, to a given data set $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$. Suppose, at some point during the execution of the EM algorithm, we get for the k^{th} component that $\boldsymbol{\mu}_k \to \mathbf{x}_n$ (for some $n \in \{1, ..., N\}$) and $\Sigma_k \to \mathbf{0}$.

(a) In this case, what would you expect the corresponding value of π_k to be? Why? **[1]**

(b) What will be the corresponding value of the likelihood $p(\mathbf{X})$? Mention the reason why. **[1]**

(c) Is this a solution that the EM algorithm could converge towards? Why or why not? **[0.5]**

(d) Would this be a desirable solution to obtain? Why or why not? If not, can you suggest some way of avoiding it? **[2]**

ELL409: Machine Intelligence and Learning

Major Test, Form: C

Maximum marks: 24

(Answer all questions on this question paper. Use the answer script only for working; it will not be graded. Read all section-specific instructions carefully.)

Name: _____

Entry Number: _____

Section 1. Multiple choice questions

Instructions: Each question may have any number of correct choices. Clearly mark (tick) all choices you believe to be correct (1 mark for each correct choice, -0.5 for each incorrect choice).

1. Which of the following are reasonable ways to select the number of clusters K for K-means?
   (a) Minimising the distortion function J as a function of K
   (b) Set a threshold τ, pick the smallest K such that $J < \tau$
   (c) Set a threshold τ, pick the smallest K such that the reduction in J from $K-1$ to K clusters is less than τ
   (d) Set a threshold τ, pick the smallest K such that the reduction in J from K to $K+1$ clusters is less than τ
2. Which of the following might be valid reasons for preferring a neural network over an SVM?
   (a) An neural net effectively applies a non-linear transformation on the input space; an SVM cannot.
   (b) The model size (number of parameters) for a neural net is fixed in advance, whereas for an SVM it depends on the number of support vectors and hence on the training data.
   (c) A neural net should not get stuck in local minima, unlike an SVM.
   (d) Neural nets extend more naturally to multi-class classification than SVMs.
   (e) Neural nets have an inbuilt regularisation mechanism to avoid overfitting, unlike SVMs.

Section 2. Numerical/Short-answer questions

Instructions: Please write *only the final answers* on this question paper, as concisely as possible, in the space provided for each item. The provided answer script should be used for all working (where applicable), but will not be graded. However, in case of any doubt regarding your answers, we may refer to the answer script to check your working. So please try to write out your working as clearly as possible.

3. Suppose you have run K-means on a labeled data set (without using the labels during clustering), containing 8 classes with 100 images from *each* class. Now, you seek to assess the accuracy of your clustering by assigning to all points in a given cluster a 'predicted label', which is taken to be the true label found most frequently within that cluster. You then report as the accuracy the overall fraction of points which are assigned to the correct label as per this labeling.

   (a) Assuming you have taken the number of clusters K to be 20 or fewer, what is the lowest possible accuracy this could give you? **[1]**

   (b) If we remove this upper bound on K, are there values of K for which this lowest possible accuracy will be greater than the above? If so, what is the smallest value of K for which this will happen, for the given data set? **[1]**

   (c) What is the smallest K at which 100% accuracy can theoretically be achieved for the given data set? **[0.5]**

(d) Why is picking K so as to maximise this notion of accuracy not a good way of determining the number of clusters? **[0.5]**

4. The idea behind gradient descent is that on each iteration, you shift the model parameters by a small amount in a direction (that of the negative gradient) so as to decrease the error. Hence, in theory the error should get lower the longer you run gradient descent for. So why is early stopping often used in training neural networks? Explain as precisely as you can. **[2]**

5. Which supervised learning or prediction technique would you use if you knew the underlying joint distribution of your features and labels, *i.e.*, $p(\mathbf{x}, t)$? Give the precise mathematical model $y(\mathbf{x})$ for it. **[2]**

6. We have seen that the hidden units in neural nets can implement any suitable non-linear activation function. One of the most widely used activation functions nowadays is the *Leaky ReLU* (Rectified Linear Unit), which we will define as follows:

$$LR(a) = \begin{cases} a & \text{if } a > 0 \\ 0.01a & \text{otherwise} \end{cases}$$

Consider a single hidden unit called z, which implements the above activation function. Take z to be connected to D input features $x_1, x_2, ..., x_D$, with corresponding weights $w_1, w_2, ..., w_D$.

(a) Draw a picture of the unit z with all its incoming connections and the corresponding input units. Also include the bias term w_0 in your picture via a dummy input variable. All units and weights should be labeled clearly. You picture should include the expression for the value of z as a function of the input units. **[1.5]**

(b) For backpropagation, we wish to calculate the error gradient with respect to each of these weights just defined, *i.e.*, $\frac{\partial E(\mathbf{w})}{\partial w_i}, i \in \{0, 1, 2, ...D\}$. Apply the chain rule using z as the intermediate variable, to write this partial derivative as a product of two partial derivatives. Where will the value of the first one come from? **[1]**

(c) Here we wish to show the calculation for the second partial derivative obtained via the above chain rule. To do this, we will first need to obtain the derivative of the *Leaky ReLU* function with respect to its argument. Write down (in an appropriate form) the value of this derivative, *i.e.*, $\frac{\partial LR(a)}{\partial a}$. Is there any value of a where it is undefined? **[2]**

(d) Now write down (again, in appropriate form) the entire expression for the value of the second partial derivative from the above chain rule. **[1.5]**

(e) Is there any specific situation in which backpropagation through a Leaky ReLU hidden unit would be theoretically ill-defined? If so, what is it, and why do you think Leaky ReLUs are still usable in practice? **[1.5]**

7. Suppose you run PCA on a high-dimensional labeled data set, and find that only a small number of principal components capture most of the variance (say $> 90\%$). Is it always safe to replace the original features with the smaller number of principal components, for the purposes of supervised learning? If yes, give a general justification. If no, explain using a simple counter-example. **[2]**

8. Consider a GMM with K mixture components being fitted via the EM algorithm for likelihood maximisation, to a given data set $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$. Suppose, at some point during the execution of the EM algorithm, we get for the k^{th} component that $\boldsymbol{\mu}_k \to \mathbf{x}_n$ (for some $n \in \{1, ..., N\}$) and $\Sigma_k \to \mathbf{0}$.

(a) In this case, what would you expect the corresponding value of π_k to be? Why? **[1]**

(b) What will be the corresponding value of the likelihood $p(\mathbf{X})$? Mention the reason why. **[1]**

(c) Is this a solution that the EM algorithm could converge towards? Why or why not? **[0.5]**

(d) Would this be a desirable solution to obtain? Why or why not? If not, can you suggest some way of avoiding it? **[2]**

ELL409: Machine Intelligence and Learning

Major Test, Form: D

Maximum marks: 24

(Answer all questions on this question paper. Use the answer script only for working; it will not be graded. Read all section-specific instructions carefully.)

Name: _____

Entry Number: _____

Section 1. Multiple choice questions

Instructions: Each question may have any number of correct choices. Clearly mark (tick) all choices you believe to be correct (1 mark for each correct choice, -0.5 for each incorrect choice).

1. Which of the following are reasonable ways to select the number of clusters K for K-means?
   (a) Minimising the distortion function J as a function of K
   (b) Set a threshold τ, pick the smallest K such that $J < \tau$
   (c) Set a threshold τ, pick the smallest K such that the reduction in J from $K-1$ to K clusters is less than τ
   (d) Set a threshold τ, pick the smallest K such that the reduction in J from K to $K+1$ clusters is less than τ
2. Which of the following might be valid reasons for preferring a neural network over an SVM?
   (a) An neural net effectively applies a non-linear transformation on the input space; an SVM cannot.
   (b) The model size (number of parameters) for a neural net is fixed in advance, whereas for an SVM it depends on the number of support vectors and hence on the training data.
   (c) A neural net should not get stuck in local minima, unlike an SVM.
   (d) Neural nets extend more naturally to multi-class classification than SVMs.
   (e) Neural nets have an inbuilt regularisation mechanism to avoid overfitting, unlike SVMs.

Section 2. Numerical/Short-answer questions

Instructions: Please write *only the final answers* on this question paper, as concisely as possible, in the space provided for each item. The provided answer script should be used for all working (where applicable), but will not be graded. However, in case of any doubt regarding your answers, we may refer to the answer script to check your working. So please try to write out your working as clearly as possible.

3. Suppose you have run K-means on a labeled data set (without using the labels during clustering), containing 10 classes with 50 images from *each* class. Now, you seek to assess the accuracy of your clustering by assigning to all points in a given cluster a 'predicted label', which is taken to be the true label found most frequently within that cluster. You then report as the accuracy the overall fraction of points which are assigned to the correct label as per this labeling.

   (a) Assuming you have taken the number of clusters K to be 20 or fewer, what is the lowest possible accuracy this could give you? **[1]**

   (b) If we remove this upper bound on K, are there values of K for which this lowest possible accuracy will be greater than the above? If so, what is the smallest value of K for which this will happen, for the given data set? **[1]**

   (c) What is the smallest K at which 100% accuracy can theoretically be achieved for the given data set? **[0.5]**

(d) Why is picking K so as to maximise this notion of accuracy not a good way of determining the number of clusters? **[0.5]**

4. The idea behind gradient descent is that on each iteration, you shift the model parameters by a small amount in a direction (that of the negative gradient) so as to decrease the error. Hence, in theory the error should get lower the longer you run gradient descent for. So why is early stopping often used in training neural networks? Explain as precisely as you can. **[2]**

5. Which supervised learning or prediction technique would you use if you knew the underlying joint distribution of your features and labels, *i.e.*, $p(\mathbf{x}, t)$? Give the precise mathematical model $y(\mathbf{x})$ for it. **[2]**

6. We have seen that the hidden units in neural nets can implement any suitable non-linear activation function. One of the most widely used activation functions nowadays is the *Leaky ReLU* (Rectified Linear Unit), which we will define as follows:

$$LR(a) = \begin{cases} a & \text{if } a > 0 \\ 0.01a & \text{otherwise} \end{cases}$$

Consider a single hidden unit called z, which implements the above activation function. Take z to be connected to D input features $x_1, x_2, ..., x_D$, with corresponding weights $w_1, w_2, ..., w_D$.

(a) Draw a picture of the unit z with all its incoming connections and the corresponding input units. Also include the bias term w_0 in your picture via a dummy input variable. All units and weights should be labeled clearly. You picture should include the expression for the value of z as a function of the input units. **[1.5]**

(b) For backpropagation, we wish to calculate the error gradient with respect to each of these weights just defined, *i.e.*, $\frac{\partial E(\mathbf{w})}{\partial w_i}, i \in \{0, 1, 2, ...D\}$. Apply the chain rule using z as the intermediate variable, to write this partial derivative as a product of two partial derivatives. Where will the value of the first one come from? **[1]**

(c) Here we wish to show the calculation for the second partial derivative obtained via the above chain rule. To do this, we will first need to obtain the derivative of the *Leaky ReLU* function with respect to its argument. Write down (in an appropriate form) the value of this derivative, *i.e.*, $\frac{\partial LR(a)}{\partial a}$. Is there any value of a where it is undefined? **[2]**

(d) Now write down (again, in appropriate form) the entire expression for the value of the second partial derivative from the above chain rule. **[1.5]**

(e) Is there any specific situation in which backpropagation through a Leaky ReLU hidden unit would be theoretically ill-defined? If so, what is it, and why do you think Leaky ReLUs are still usable in practice? **[1.5]**

7. Suppose you run PCA on a high-dimensional labeled data set, and find that only a small number of principal components capture most of the variance (say $> 90\%$). Is it always safe to replace the original features with the smaller number of principal components, for the purposes of supervised learning? If yes, give a general justification. If no, explain using a simple counter-example. **[2]**

8. Consider a GMM with K mixture components being fitted via the EM algorithm for likelihood maximisation, to a given data set $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$. Suppose, at some point during the execution of the EM algorithm, we get for the k^{th} component that $\boldsymbol{\mu}_k \to \mathbf{x}_n$ (for some $n \in \{1, ..., N\}$) and $\Sigma_k \to \mathbf{0}$.

(a) In this case, what would you expect the corresponding value of π_k to be? Why? **[1]**

(b) What will be the corresponding value of the likelihood $p(\mathbf{X})$? Mention the reason why. **[1]**

(c) Is this a solution that the EM algorithm could converge towards? Why or why not? **[0.5]**

(d) Would this be a desirable solution to obtain? Why or why not? If not, can you suggest some way of avoiding it? **[2]**