What is this exercise sheet for?

At the first lab session you will have encountered the following:

1. For p-dimensional vectors \( \mathbf{w}, \mathbf{x} \in \mathbb{R}^p \) linear expressions \( y_n = w_0 + \mathbf{x}_n^T \mathbf{w} \) are written as dot products

\[
y_n = (1 \ x_n^T) \begin{pmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_p \end{pmatrix} = (1 \ x_{n,1} \ x_{n,2} \ldots \ x_{n,p}) \begin{pmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_p \end{pmatrix}, \quad n = 1, \ldots, N,
\]

2. Each such expression for \( n = 1, \ldots, N \) is placed in a row of the design matrix,

3. Gradients of loss functions coded up as matrix vector products.

These exercises will help you work through the steps of performing derivatives of functions of matrices and vectors that will routinely come up in all the methods and algorithms you will study in this module. This is the kind of mathematical language you are expected to become comfortable with by the middle of the semester.

Partial derivatives and matrix calculus

1. Using the symbol \( \delta_{ab} \), the Kronecker delta

\[
\delta_{ab} = \begin{cases} 
1, & a = b \\
0, & a \neq b
\end{cases}
\]

show the following:
1. for vector \( v \) with components \( v_i \), \( \sum_i v_i \delta_{ij} = v_j \);
   (b) for matrix \( A \) with elements \((A)_{ij} = a_{ij}\), \( \sum_j a_{ij} \delta_{jk} = a_{ik} \);
   (c) for matrices \( A, B \), the element of the \( i^{th} \) diagonal of \( C = AB \) is expressed as \( \sum_k a_{ij} b_{jk} \delta_{ki} \);
   (d) the trace of a matrix is \( \text{tr}(A) = \sum_{ij} a_{ij} \delta_{ij} \)
   (e) \( \frac{\partial w_a}{\partial w_b} = ab \).

2. For \( p \times p \) matrix \( A \) with matrix elements \((A)_{ij} = a_{ij} \), \( 1 \leq i, j \leq p \) and vector \( x = (x_1, \ldots, x_p)^T \) show that:
   (a) the \( i \)-th element of vector \((Ax)\) is \((Ax)_i = \sum_j a_{ij} x_j \);
   (b) \( \nabla_x (Ax) := \frac{\partial}{\partial x} (Ax) = A^T \). Write out the indices explicitly:
   \[
   \left( \frac{\partial}{\partial x} (Ax) \right)_{ij} = \frac{\partial}{\partial x_i} (Ax)_j = \frac{\partial}{\partial x_i} \sum_k a_{jk} x_k ;
   \]
   (c) the gradient of the scalar quadratic form \( x^T Ax \) is
   \[
   \nabla_x (x^T Ax) = (A + A^T)x ;
   \]
   hint: the \( i \)-th matrix element of the gradient is
   \[
   \frac{\partial}{\partial x_i} \sum_{pq} x_p a_{pq} x_q ;
   \]
   (d) the partial derivative of the quadratic form \( x^T Ax \) with respect to \( A \) can be evaluated for each matrix element \( a_{ij} \), \( 1 \leq i, j \leq p \):
   \[
   \frac{\partial}{\partial a_{ij}} \left( \sum_{rs} x_r a_{rs} x_s \right) \]
   with the result
   \[
   \nabla_A (x^T Ax) = xx^T ;
   \]
   with \( xx^T \) a \( p \times p \) matrix.
3 Weight updates in linear regression

In some cases, involving linear combinations of functions of the data:

\[ \hat{y}_n = f(x_n, w) = \sum_{j=1}^{p} w_j \phi_j(x_n) \]

the vanishing of the gradient of the quadratic loss function can lead to an analytical expression for the best-fit weights \( w = (w_1, \ldots, w_p) \). In what follows, we set \( \phi_j(x_n) = x_n \) but the same mathematical steps hold straightforwardly if a different set of functions are used to map the input data.

**Exercise:** (linearity) Show that \( y(x; w) := w_0 + w_1 x + w_2 x^2 \) is linear in \( w \).

**Hint:** verify that \( y(x; aw + a'w') = ay(x, w) + a'y(x, w') \), where \( a, a' \in \mathbb{R} \).

**Example:** We could fit a quadratic function \( y(x; w) := w_0 + w_1 x + w_2 x^2 \) to a set of points \( D = \{(x_1, y_1), \ldots, (x_N, y_N)\} \) by minimising the loss function

\[ \frac{1}{N} \sum_{n=1}^{N} (y_n - (w_0 + w_1 x_n + w_2 x_n^2))^2. \]

This would also fall under the heading of linear regression as the weights appear linearly in the function being fitted to data.

3.1 Gradient of squared loss

The squared residual \( l(w, x_n, y_n) \) for each data point \((x_n, y_n)\) is \((\hat{y}_n - y_n)^2\), where \( \hat{y}_n = w^T x_n = w_0 + \sum_{p=1}^{p} w_p x_{n,j} \), where \( x_{n,j} \) is the \( j \)th component of the \( p \)-dimensional input vector \( x_n \). The loss function to be minimised is the average over the sample data set

\[ L(w) := E_{(x_n, y_n) \sim D} [l(w, x_n, y_n)] = \frac{1}{N} \sum_{n=1}^{N} (y_n - w^T x_n)^2. \]

The components of the gradient \( (\nabla_w L(w))_k \) are

\[ \frac{\partial}{\partial w_k} \left( \sum_{n} (y_n - \sum_{p} w_p x_{n,p}) (y_n - \sum_{q} w_p x_{n,q}) \right). \]

In the same way as in the exercises above, the partial derivatives as evaluated thus:

\[ \sum_{n} (y_n - w^T x_n) \left( -\sum_{p} x_{n,p} \delta_{pk} - \sum_{q} x_{n,q} \delta_{qk} \right) = -2 \sum_{n} (y_n - w^T x_n)(x_{n,k}) \]
Exercise: Compare the result to the weight updates in the gradient descent routines in the lab exercises (Jupyter notebook).

3.2 Minimising squared loss leads to normal equation

To show: The components of the gradient \( (\nabla_w L(w))_k \) are set to 0 yielding

\[
X^T y = (X^T X) w \implies w = (X^T X)^{-1} X^T y. \quad (i)
\]

Rearranging terms in \(-2 \sum_n (y_n - w^T x_n)(x_{n,k}) = 0\) from above,

\[
\sum_{n=1}^N y_n x_{n,k} = \sum_{n=1}^N w^T x_n x_{n,k} = \sum_{j=1}^p \sum_{n=1}^N x_{n,k} x_{n,j} w_j = \sum_{j=1}^p w_j \sum_{n=1}^N x_{n,k} x_{n,j} = \sum_{j=1}^p (X^T)_k (X)_n w_j
\]

and the result (i) follows.

3.3 The normal equation: residual normal to range of matrix

For a matrix equation \( Xw = y \), the residual \( r = (y - Xw) \) is orthogonal to the range of \( X \). Show that the condition for all the columns of \( X \) to be orthogonal to \( r \) reduces to an equation of the form (i).

(Moore-Penrose) pseudo-inverse \( A^+ \) of a matrix \( A \) is defined as follows:

\[
A^+ := (A^T A)^{-1} A^T , \quad \text{satisfies } AA^+ A = A,
\]

in terms of which the solutions \( z \in \mathbb{R}^n \) of a general linear equation \( Ax = b \) can be written as

\[
z = A^+ b + (I - A^+ A) v,
\]

for an arbitrary \( v \in \mathbb{R}^n \). Verify that \( z = A^+ b \) is a particular solution to \( Ax = b \) and \( u \triangleq (I - A^+ A) v \) belongs to the kernel of \( A \), \( u \in \ker(A) \).

4 Binary classification: introducing cost functions

4.1 Misclassification count: max loss

Linear classifier: given data \( \mathcal{D} = ((x_1, y_1), \ldots, (x_n, y_n), (x_N, y_N)) \) where \( x_n \in \mathbb{R}^p \) is a \( p \)-dimensional feature vector and \( y_n \in \{-1, +1\} \) a binary label, learn weights \( w_0, w \) so that

\[
\hat{y}_n = \text{sign}(w_0 + w^T x_n).
\]
For a correct classification $\hat{y}_n$ and $y_n$ agree, so $\hat{y}_n y_n = 1 > 0$, else $\hat{y}_n y_n = -1 < 0$.

Show that the number of misclassified points is

$$\sum_{n=1}^{N} \max (0, -\hat{y}_n y_n) = \sum_{n=1}^{N} \max (0, -(w_0 + w^T x_n) y_n)$$

This is called, variously, a max cost, a hinge loss or rectified linear unit (ReLU) loss function.

### 4.2 Partial derivatives of composition of functions

If a function $h : \mathbb{R} \to \mathbb{R}$ that maps $x$ to $h(x)$ has a derivative $h'(x)|_{x=a}$ that is defined at $x = a$ and if there is a function $g : \mathbb{R} \to \mathbb{R}$ that maps $h = h(x)$ to $g(h)$ such that the derivative $g'(h)|_{h=h(a)}$ exists, then

$$f'(x)|_{x=a} = (g \circ h)'(a) = (g'(a))h'(a).$$

Expressed another way, if $f(x) = g(h(x))$ then

$$\frac{df(x)}{dx} \bigg|_{x=a} = \frac{dg(h)}{dh} \frac{dh(x)}{dx} \bigg|_{x=a} = g'(h(a))h'(a).$$

**Example:** If $h(x) = \sin(x)$ and $g(x) = e^{-\frac{x^2}{2}}$, then $f(x) = g(h(x)) = e^{-\frac{\sin^2 x}{2}}$ and

$$\frac{df(x)}{dx} = \frac{de^{-\frac{h^2}{2}}}{dh} \frac{d\sin(x)}{dx} = -h(x)e^{-\frac{h^2}{2}} \cos(x) = -\sin x \cos x e^{-\frac{\sin^2 x}{2}}.$$

### 4.3 Logistic regression: gradient of softmax loss

In the lecture slides we showed that the softmax loss function can be expressed in terms of the extended weights $\bar{w}$ and extended data points $\bar{x}_n$:

$$\sum_{n} \ln \left( 1 + \exp \left( -y_n \begin{bmatrix} w_0 \\ w \end{bmatrix}^T \begin{bmatrix} 1 \\ x_n \end{bmatrix} \right) \right) = \sum_{n} \ln (1 + \exp (-y_n \bar{w}^T \bar{x}_n)) \equiv L(\bar{w})$$

1. Verify, using the definition $\sigma(z) = (1 + \exp(-z))^{-1}$,

$$\frac{d}{dw} \ln(1 + e^{-wx}) = -x \sigma(-wx),$$

5
2. and
\[ \frac{d}{dw_1} \ln(1 + e^{-w_1 x_1 - w_2 x_2}) = -x_1 \sigma(-w_1 x_1 - w_2 x_2). \]

3. Using the above, show
\[ (\nabla w)_i L(\hat{w}) = - \sum_{n=1}^{N} y_n \tilde{x}_{n,i} \sigma \left( -y_n \hat{w}^T \tilde{x}_n \right), \]
where \( \tilde{x}_{n,i} \) is the \( i \text{th} \) component (feature) of the \( n \text{th} \) data point \( x_n \).