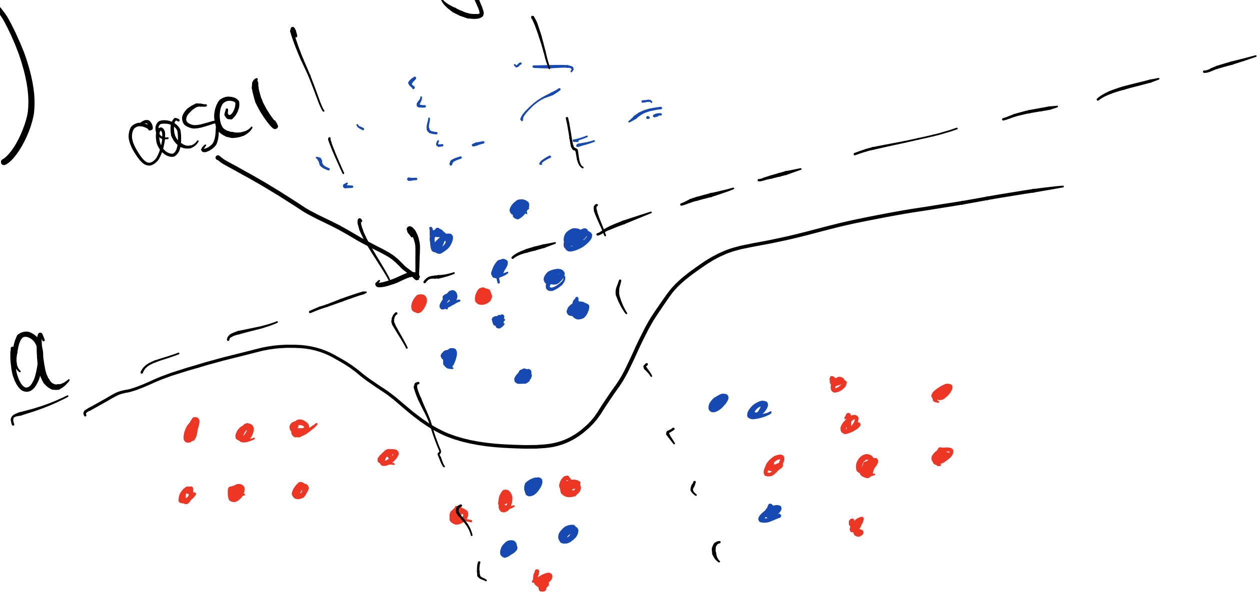
1.

b) Since we're using just one hidden layer, and one hidden unit, this neural net is similar to logistic regression. Thus, we get similar decision boundary.

1.f) Since we have more hidden units NN can separate the data non-linearly, so the model get a better decision boundaries. It happens because now we have more weights and there diffrent ways how they can be adjusted.

1j)



a gives the boundary for the cases like case 1 to define the class.

1L) area : 0.93

AUC is used as a single performance measure for precision-recall curves.
Since our model is better (more accurate) than in A2, the area is greater.

3d) Since sgd computing gradient only using a piece (150 data points) of data, it's much faster.
We still get a good approximation of gradient, but we perform much less updates of the weights (epochs)

(2)

a) Prove $\frac{\partial c(t,o)}{\partial W_j} = (o-t)h_j$

$c(t,o) = -t\log(o) - (1-t)\log(1-o)$

$= -t\log\left(\frac{1}{1+\exp(-z)}\right) - (1-t)\log\left(\frac{\exp(-z)}{1+\exp(-z)}\right) =$

$= t\log(1+\exp(-z)) + (1-t)(z+\log(1+\exp(-z)))$

$= \cancel{t\log(1+\exp(-z))} + z + \log(1+\exp(-z)) - tz - \cancel{t\log(1+\exp(-z))}$

$= z + \log(1+\exp(-z)) - tz = \log(1+\exp(-z)) + z(1-t)$

$$\frac{\partial c(t,o)}{\partial W_j} = \frac{\partial(1+\exp(-z))}{\partial W_j} + \frac{\partial(z(1-t))}{\partial W_j} =$$

$$= \left(-\frac{\partial z}{\partial W_j} \cdot \frac{-\exp(-z)}{1+\exp(-z)} + \frac{\partial z}{\partial W_j}(1-t)\right) =$$

$$= \frac{\partial z}{\partial W_j}\left(\frac{-\exp(-z)}{1+\exp(-z)} + (1-t)\right) =$$

$$= \frac{\partial z}{\partial W_j}(-1+o+1-t) = \frac{\partial z}{\partial W_j}(o-t) =$$

$= h_j(o-t)$ as needed

b) $c(t,o) = -\log(1+\exp(-z)) + z(1-t)$

$$\frac{\partial c(t,o)}{\partial V_{ij}} = \frac{\partial(-\log(1+\exp(-z)))}{\partial V_{ij}} + \frac{\partial z(1-t)}{\partial V_{ij}} =$$

$$z = \tanh(x_i V + v_0) W_j + w_0$$

$$= \left(\frac{\partial z}{\partial V_{ij}} \cdot \frac{\exp(-z)}{1+\exp(-z)} + \frac{\partial z}{\partial V_{ij}}(1-t)\right) =$$

$$= \frac{\partial z}{\partial V_{ij}}\left(\frac{\exp(-z)}{1+\exp(-z)} + (1-t)\right) = \frac{\partial z}{\partial V_{ij}}(-1+o+1-t) =$$

$$= \frac{\partial z}{\partial V_{ij}}^{*}(o-t) = \frac{(o-t)\, x_i w_j}{\cosh^2(u_j)}$$

$$^{*}\frac{\partial z}{\partial V_{ij}} = \frac{\partial(\tanh(xV+v_0) w_j + w_0)}{\partial V_{ij}} =$$

$$= \frac{\partial(\tanh(x_i V + v_0) w_j)}{\partial V_{ij}} = \frac{x_i w_j}{\cosh^2(u_j)}$$

(c) $c(t,0) = -\log(1+\exp(-z)) + z(1-t)$

$$\frac{\partial c(t,0)}{\partial w_0} = \frac{\partial z}{\partial w_0}(0-t) = (0-t)$$

$$\frac{\partial(h_j w_j + w_0)}{\partial w_0} = 1$$

(d) $c(t,0) = -\log(1+\exp(-z)) + z(1-t)$

$$\frac{\partial c(t,0)}{\partial v_j} \overset{*}{=} \frac{\partial z}{\partial v_j}(0-t) = \frac{w_j}{\cosh^2(u_j)}(0-t)$$

$$^{*}\frac{\partial z}{\partial v_j} = \frac{\partial(\tanh(u_j)\overset{z}{\overset{\shortparallel}{w_j}} + w_0)}{\partial v_j} =$$

$$= \left(\frac{\partial u_j}{\partial v_j} \cdot \frac{w_j}{\cosh^2(u_j)}\right) = \frac{\partial(x_i v_j + v_0)}{\partial v_j} \cdot \frac{w_j}{\cosh^2(u_j)} =$$

$$= \frac{w_j}{\cosh^2(u_j)}$$

(e) $C = c(t^1, o^1) + c(t^2, o^2) + c(t^3, o^3) \dots + c(t^n, o^n)$

$$\frac{\partial C}{\partial W} = \frac{\partial C}{\partial \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}} =$$

$$= \begin{bmatrix} \frac{\partial C}{\partial w_1} \\ \frac{\partial C}{\partial w_2} \\ \frac{\partial C}{\partial w_3} \end{bmatrix} = \begin{bmatrix} \frac{\partial \left(c(t^{(1)}, o^{(1)}) + \dots + c(t^{(n)}, o^{(n)}) \right)}{\partial w_1} \\ \vdots \\ \frac{\partial \left(c(t^{(1)}, o^{(1)}) + \dots + c(t^{(n)}, o^{(n)}) \right)}{\partial w_3} \end{bmatrix} =$$

$$= \begin{bmatrix} (o^{(1)} - t^{(1)}) h_{11} + (o^{(2)} - t^{(2)}) h_{12} + \dots (o^{(n)} - t^{(n)}) h_{1n} \\ \vdots \qquad\qquad \vdots \qquad\qquad \vdots \\ (o^{(1)} - t^{(1)}) h_{31} + (o^{(2)} - t^{(2)}) h_{32} + \dots (o^{(n)} - t^{(n)}) h_{3n} \end{bmatrix}$$

$$= \begin{bmatrix} h_{11} & h_{12} & h_{13} & \cdots & h_{1n} \\ h_{21} & h_{22} & h_{23} & \cdots & \\ h_{31} & h_{32} & h_{33} & \cdots & h_{nn} \end{bmatrix}^T \begin{bmatrix} o^{(1)} - t^{(1)} \\ o^{(2)} - t^{(2)} \\ o^{(3)} - t^{(3)} \\ \vdots \\ o^{(n)} - t^{(n)} \end{bmatrix} =$$

$$= H^T(O-T) = \frac{\partial C}{\partial W}$$

f) $\frac{\partial C}{\partial V} = X^T \cdot \underline{dU} = X^T((O-T)W^T * (1-H^2))$

from previous results:

$$dZ = O - T$$

By back-propagation:

multiple each element (not matmul)

$$dU = dZ\, W^T * \underbrace{(1-H^2)}_{} \quad * \quad \frac{1}{\cosh^2 U}$$

* Comes from (b)

$$\frac{1}{\cosh^2 U} = \operatorname{sech}^2 U = [1 - \tanh^2 U] = 1 - H^2$$

g) $\frac{dC}{dV_0}$ = np.sum(dU ,axis=0)

h) $\frac{d\hat{C}}{dW_0} = \frac{d\left(c(t^1,o^1) + c(t^2,o^2) + \dots + c(t^n,o^n)\right)}{\partial W_0} =$

from (C) we know : $\frac{\partial c(t,o)}{\partial W_0} = (o-t)$

$= \left(o^1 - t^1\right) + \left(o^2 - t^2\right) + \dots + \left(o^n - t^n\right) =$

$= \sum_{n=1}^{N} (o^n - t^n)$ can be written in vector form, so np.sum(O − T, axis=0)