Programming Assignment Overview: Planar Data Classification with one Hidden Layer

Learning objectives:

- Implement a 2-class classification NN w/ 1 HL
- Use units w/ a non-linear AF (e.g. tanh)
- Compute cross entropy loss
- Implement for. & backprop.

Contents:

1. Packages
2. Dataset
3. Simple log. reg.
4. NN model
   - 4.1. Defining the NN structure
   - 4.2. Initialise the model's parameters
   - 4.3. Loop

5. Performance on other datasets

FP: Example $x^{(i)}$:

$$z^{[1](i)} = W^{[1]}x^{(i)} + b^{[1]}$$

$$a^{[1](i)} = \tanh(z^{[1](i)})$$

$$z^{[2](i)} = W^{[2]}a^{[1](i)} + b^{[2]}$$

$$\hat{y}^{(i)} = a^{[2](i)} = \sigma(z^{[2](i)})$$

→ np.dot(,)

$$y^{(i)}_{prediction} = \begin{cases} 1 & \text{if } a^{[2](i)} > 0.5 \\ 0 & \text{otherwise} \end{cases}$$

Given predictions on all examples, cost:

$$J = -\frac{1}{m}\sum_{i=0}^{m}\left(y^{(i)}\log(a^{[2](i)}) + (1-y^{(i)})\log(1-a^{[2](i)})\right)$$

   - 4.4. Integrate 4.1, 4.2, 4.3 in nn_model()
   - 4.5. Predictions
   - 4.6. Tuning hidden layer size
     - Larger models (more HUs) can fit the train. set better, until they overfit
     - n_h=5 seems to be the best