1. You are working for a fast food chain who has asked you to build a model to predict the number of guests using the drive-through on a given date. Loading the data as a pandas dataframe and displaying the info gives:

```
#   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
0   Franchise number           20000 non-null   int64
1   City                       19758 non-null   object
2   Date                       20000 non-null   object
3   Number of guests           20000 non-null   int64
4   Temperature                19846 non-null   float64
```

(a) (3 points) How would you encode the `City` column as a numeric value? Justify your answer.

> **Solution:** One-hot encoding, because there is no natural order to impose on cities.

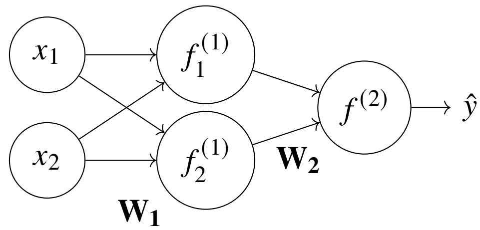
(b) (3 points) The `Temperature` column has some `null` values. Suggest a reasonable approach to deal with them.

> **Solution:** Since there aren't many features, nor are there many missing from temperature, you probably want to impute the values with something like a Nearest Neighbour imputer. This would be a reasonable choice, as the same city on a similar date would likely have a similar temperature. You could also choose just a simple median or constant imputer.

(c) (3 points) In preprocessing your data, you have chosen to normalize the numeric features. Why is it a problem to recompute the normalization parameters during inference?

> **Solution:** This would make the predicted value of a sample change depending on the other values in the inference batch. In the extreme case (inference on a single sample), you would end up with a divide by 0 error, or just normalizing to a constant.

2. Consider a simple neural network with one hidden layer as shown:



(a) (4 points) Assume that the loss function is given as $\mathcal{L}(y, \hat{y}) = \frac{1}{2}(\hat{y} - y)^2$ and $f^{(2)}(x) = x$ such that $\hat{y} = \mathbf{z}^T \mathbf{W_2}$, where $\mathbf{z}$ is the output of the hidden layer. Given the following values:

$$y = 5, \quad \hat{y} = 4, \quad \mathbf{W_2} = \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix}, \quad \mathbf{z} = \begin{bmatrix} 0.5 \\ 0.6 \end{bmatrix}$$

calculate the gradient of the loss with respect to $\mathbf{W_2}$, given as:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W_2}} = \frac{\partial \mathcal{L}}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \mathbf{W_2}}$$

Solution: It seems complicated, but this reduces down a lot! $\frac{\partial \mathcal{L}}{\partial \hat{y}} = \hat{y} - y = (4 - 5) = -1$, and $\frac{\partial \hat{y}}{\partial \mathbf{W_2}}$ is just $\mathbf{z}$. The gradient of the weight is then simply $-\mathbf{z} = \begin{bmatrix} -0.5 \\ -0.6 \end{bmatrix}$.

(b) (4 points) The previous question was calculated for a single sample. Complete the table below for the dimensions of the terms with a batch size of 8.

Term	Single Sample	Batch of 8
y	scalar	8×1
$\hat{y}$	scalar	8×1
$\mathbf{W_2}$	2×1	2×1
$\mathbf{z}$	2×1	2×8

(c) (1 point) What additional term(s) is missing or assumed to be 0 in this network?

Solution: The bias terms.