

Professor's solution to Polar Express (09/01/2023 project)

Although this problem is technically solvable using linear regression with 300 parameters, this technique would converge extremely slowly. Preprocessing input parameters can help speed it up. Here is an example of a network that works well enough:

```
def gen_smodel():
    in1=Input(shape=(1))
    in2=Input(shape=(1))
    d=tf.keras.layers.concatenate([in1,in2],axis=1)
    d = Dense(4, activation='swish')(d)
    d = Dense(2, activation = 'selu')(d)
    out=Dense(100,activation='softmax')(d)
    return Model([in1,in2],out)
```

If you have a generator available, using it will prevent overfitting.

Since the input size is quite small, it was recommended to use a big `batch_size`.

Using a lower learning rate could actually also speed up the learning process.