

## Feedback del prof su Polar Express (progetto 09-13/01/2023)

Il problema può essere essenzialmente risolto mediante regressione lineare, quindi sarebbero stati sufficienti 300 parametri. Tuttavia la convergenza è molto lenta. Un po' di processing dei parametri di input aiuta; ad esempio questa rete funziona abbastanza bene.

```
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)
```

Anche reti leggermente più grandi, con 3 o 4 neuroni sul penultimo livello sono state valutate in modo positivo.

Se avete a disposizione un generatore, è bene sfruttarlo durante il training, perchè evita problemi di overfitting.

Visto che la dimensione dei dati era piccola risultava conveniente lavorare con una `batch_size` abbastanza alta.

Anche la diminuzione del learning rate durante il training poteva velocizzare l'apprendimento.