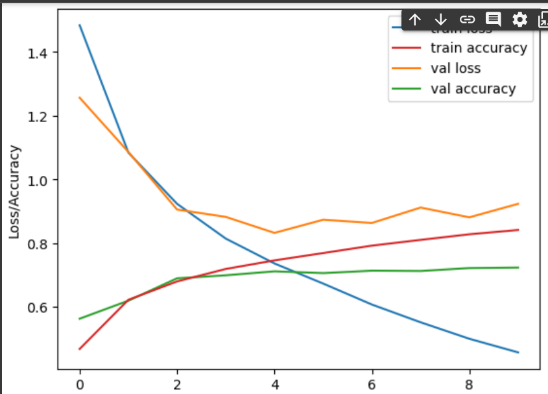
**Initial Model (no optimizations):**

- just normal data preprocessing (batch size = 64)

- ConvModel: 3 hidden layers w\ each a convolutional layer (filter size = 32, 64, 128;

activation = relu) & maxpooling + a dense output layer (activation = softmax)

 => 115,274 parameters

- optimizer = adam, loss function = categorical crossentropy

- performance after 10 epochs:

initial train loss: 1.4842, acc: 0.4670

initial val loss: 1.2565, acc: 0.5620

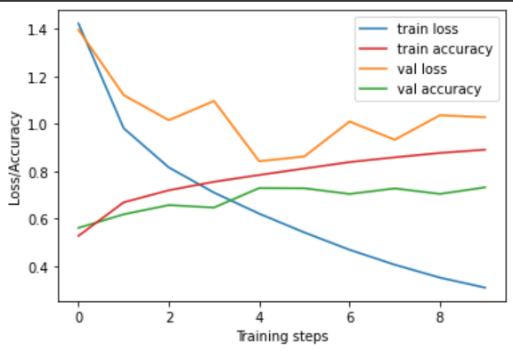
final train loss: 0.4562, acc: 0.8409

final val loss: 0.9230, acc: 0.7229

=> train and val loss at first very close (underfitting?),

then grow far apart (overfitting?), val loss & acc doesn’t

really improve after 2 epochs



**Normalization:**

- add batchNorm after each Conv layer

=> 116,170 parameters

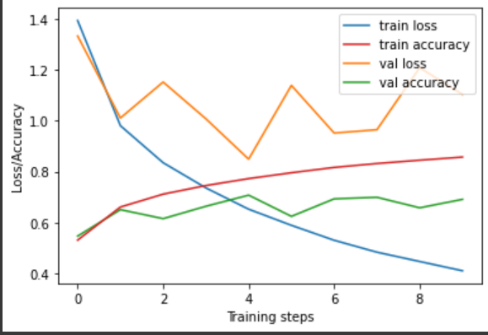
- performance after 10 epochs:

final train loss: 0.3506, acc: 0.8906

final val loss: 1.0276, acc: 0.7316

=> helped underfitting?

**Augmentation:**

**-** double the training data by flipped, cropped and

Colour altered train dataset

- performance after 10 epochs:

final train loss: 0.4098, acc: 0.8569

final val loss: 1.1033, acc: 0.6906

=>

dropout

kernel regulization

kernel initializer?

batch size

smoothing

fully connected fourth layer