Particle Tracking with Convolutional Networks

Dustin Anderson

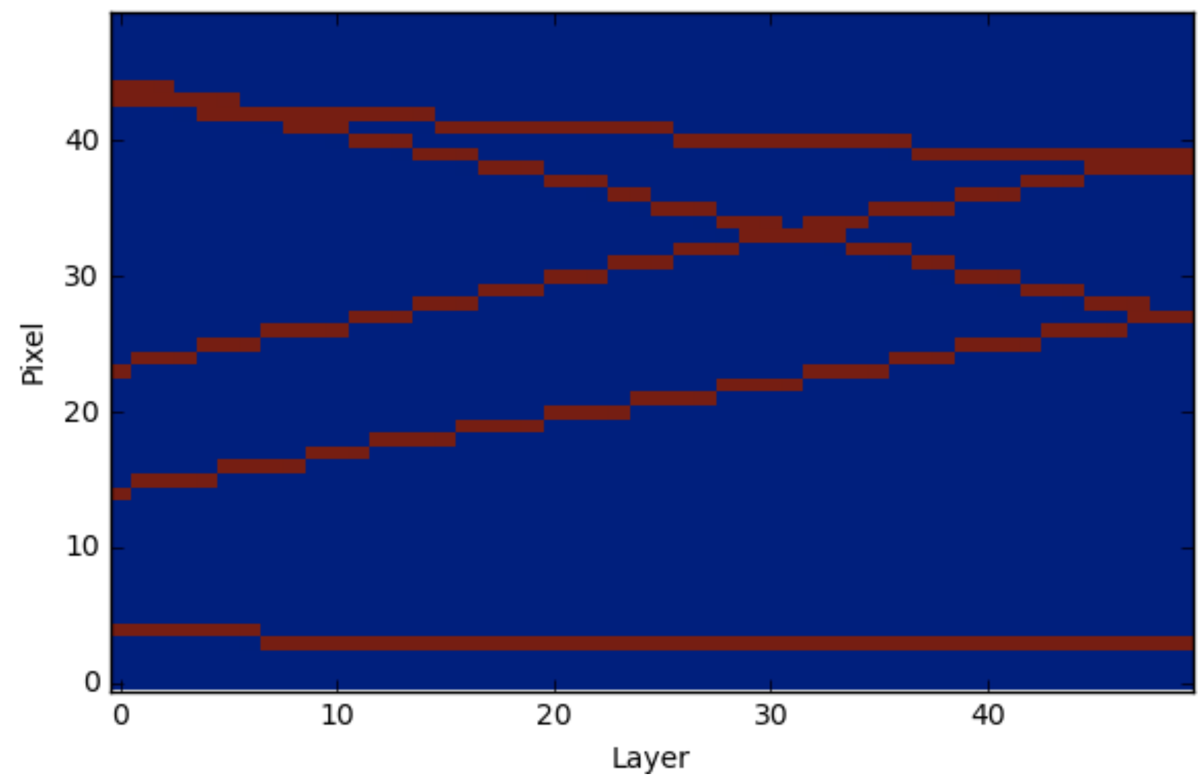
HEP.TrkX Meeting, 19 Dec. 2016

Overview

- Steve investigated an LSTM that makes layer-by-layer track predictions for a toy detector

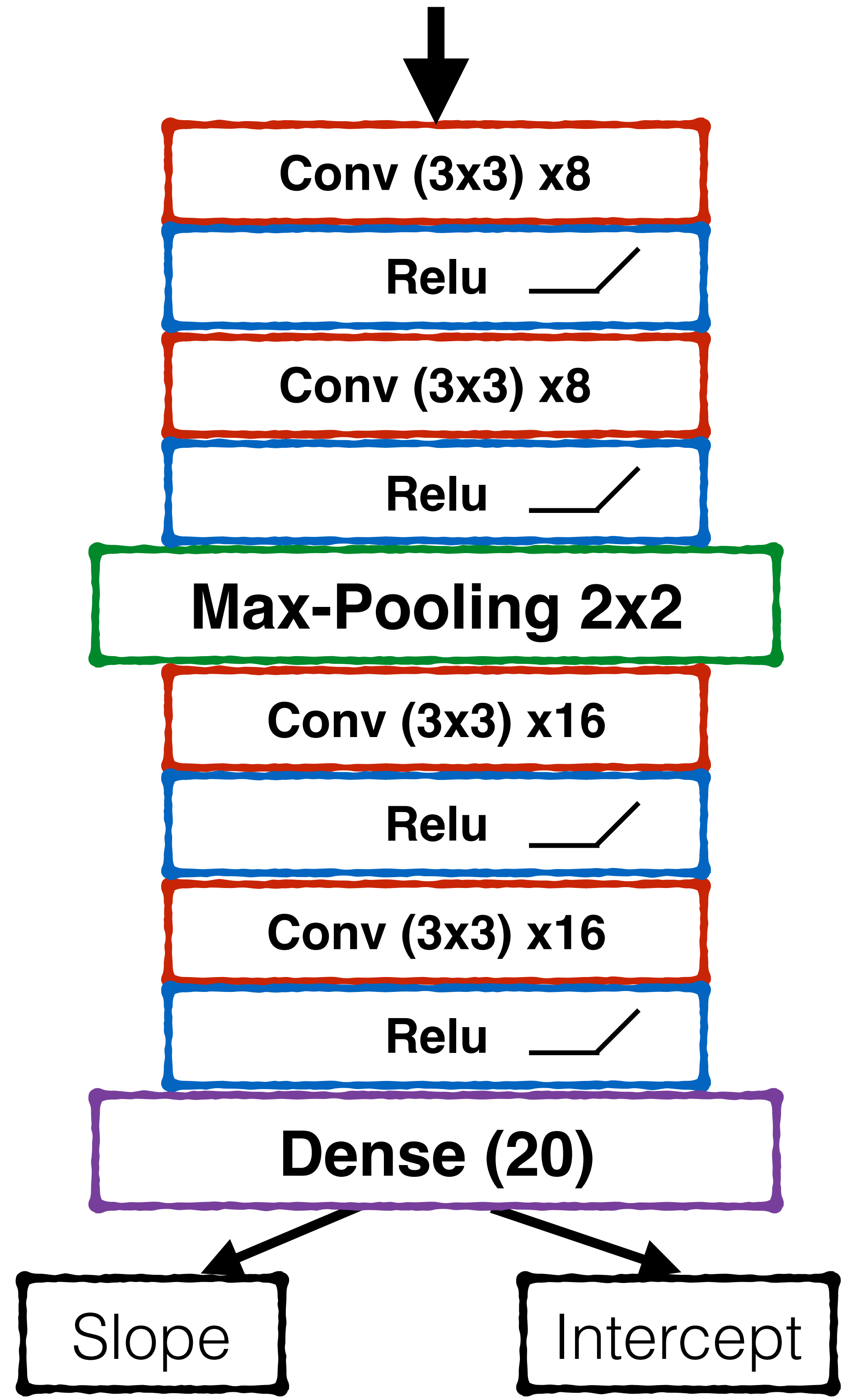
- I tried a different approach: use a convolutional net to **extract track parameters directly**

- Used Steve's detector model: 50x50 square with straight tracks



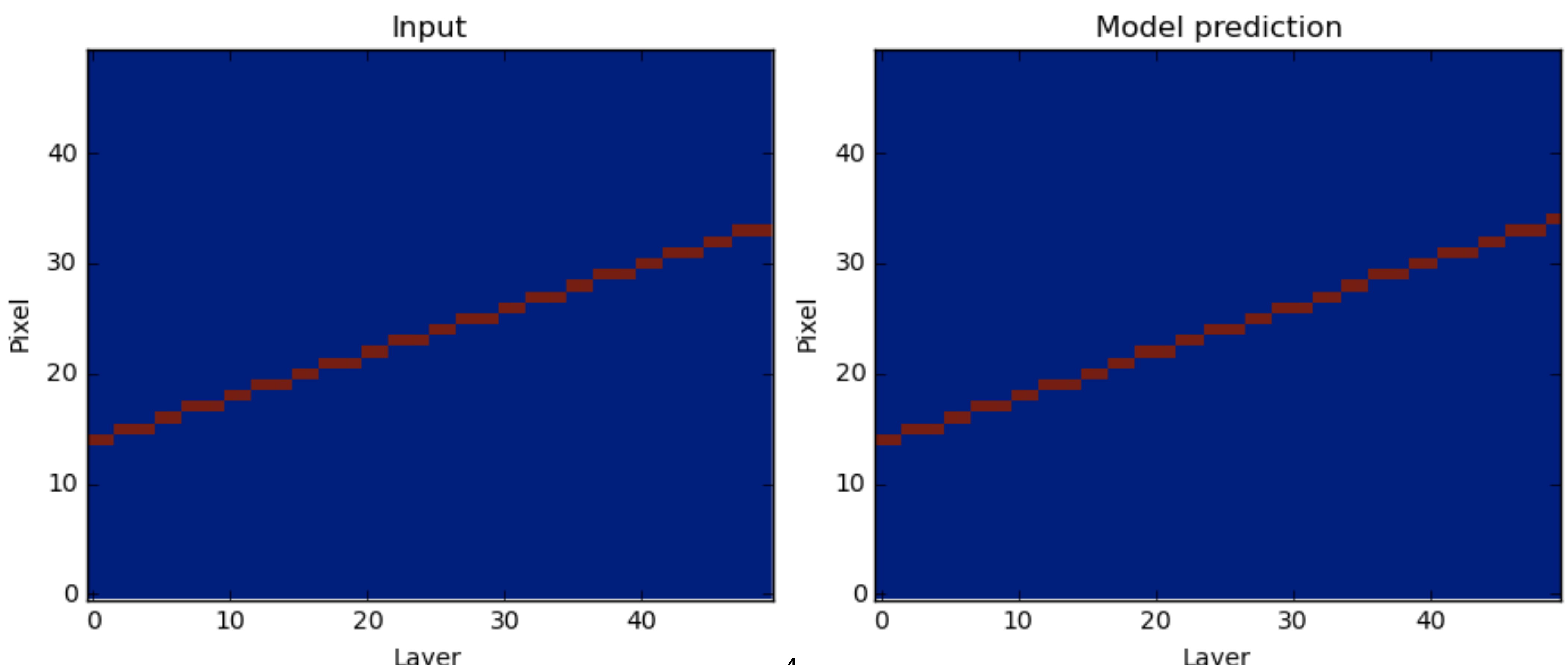
First Model

- Four convolutional layers with relu activations
- 2x2 max-pooling after the second conv. layer
- Dense layer with 20 neurons
- Output: track slope and intercept



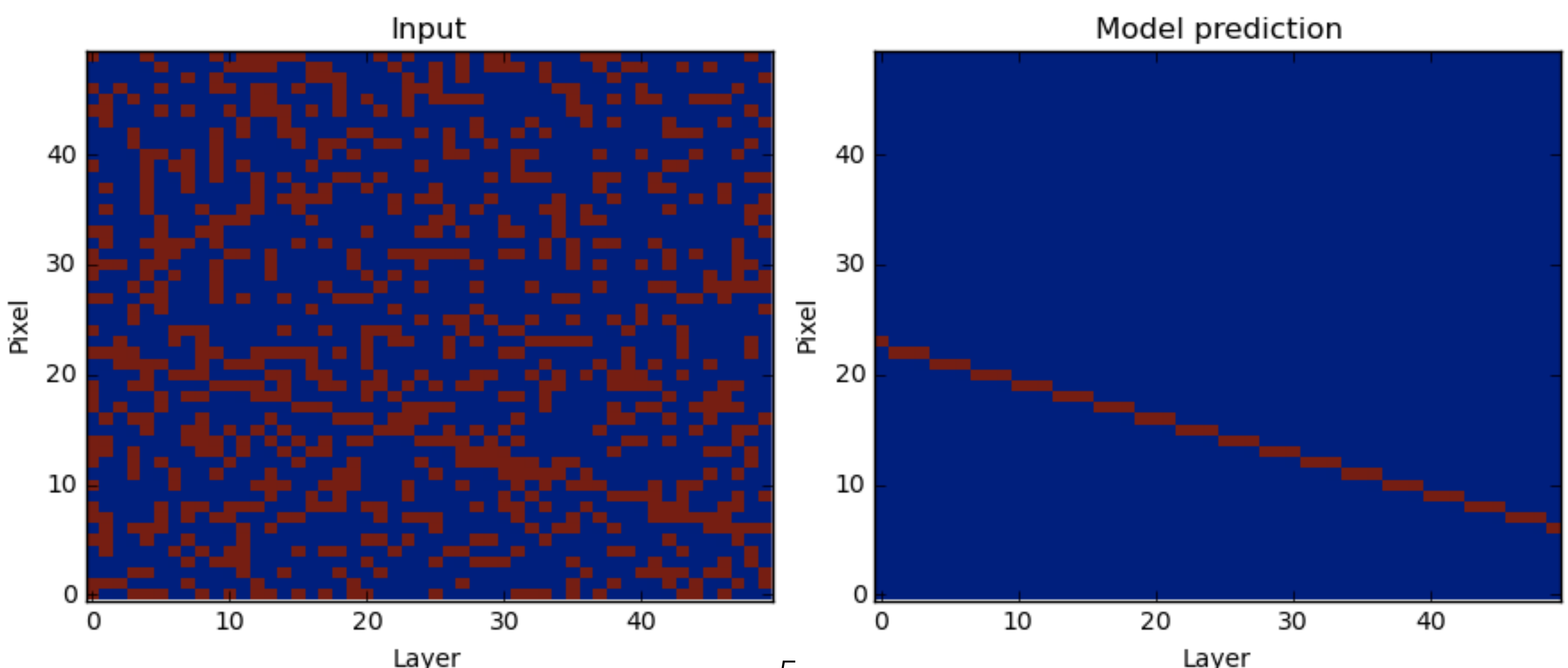
1-Track Events

- Produce training data on the fly using a generator
- Train model to predict track slope and intercept of 1 track
- Excellent performance out of the box after 2.5M training events (average MSE loss: 0.017)



1-Track Events

- **Inject noise** into the events: each pixel has 30% probability to fire
- Train on 1-track events with noise
- The model still predicts track parameters reliably

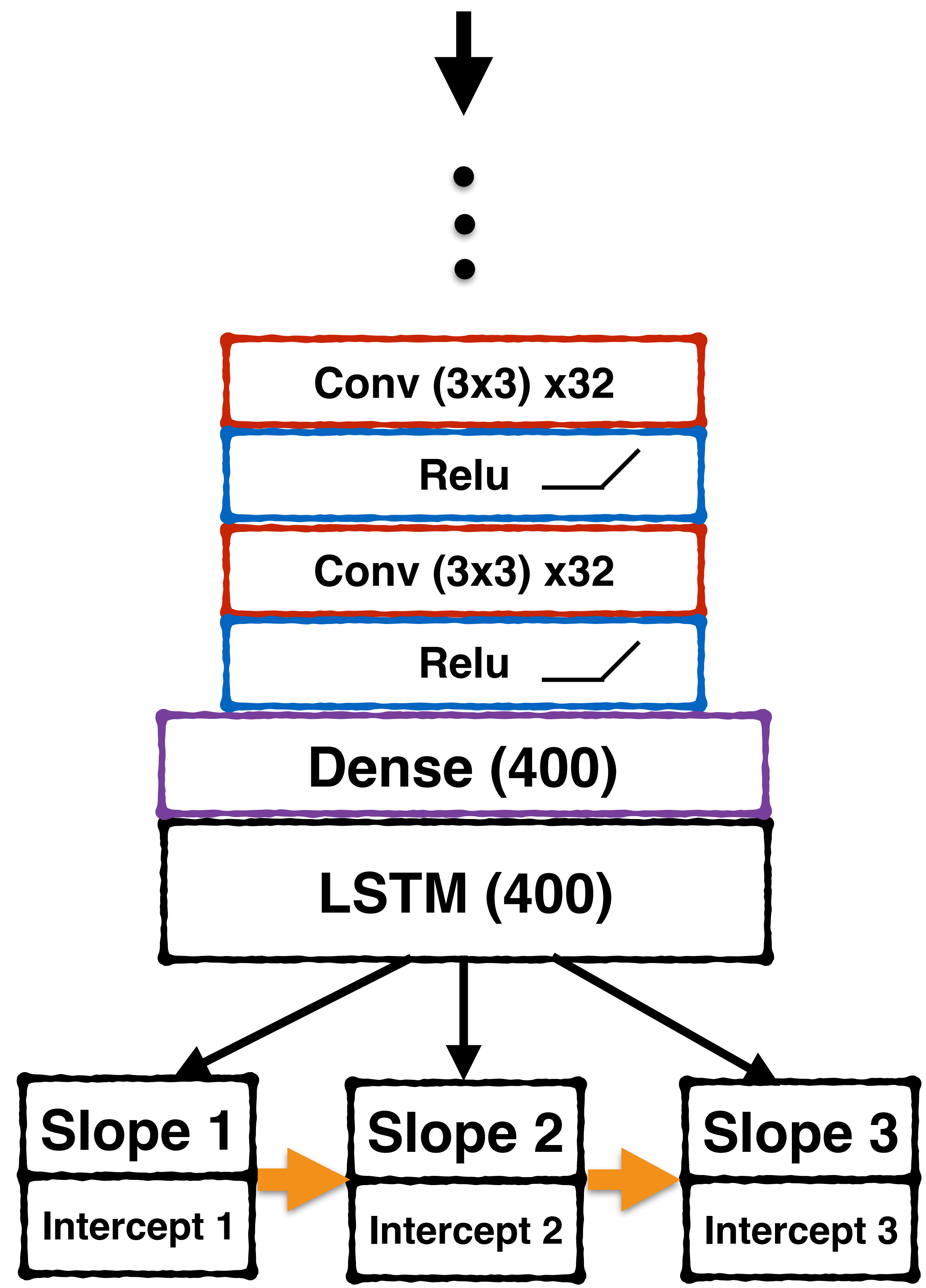


3-Track Events

- Next: move to events with N tracks (start with N=3)
- Deal with multiple tracks by **stacking an LSTM on top of the single-track model architecture**
- “Turn the crank” on the LSTM three times to output parameters for tracks 1, 2, 3

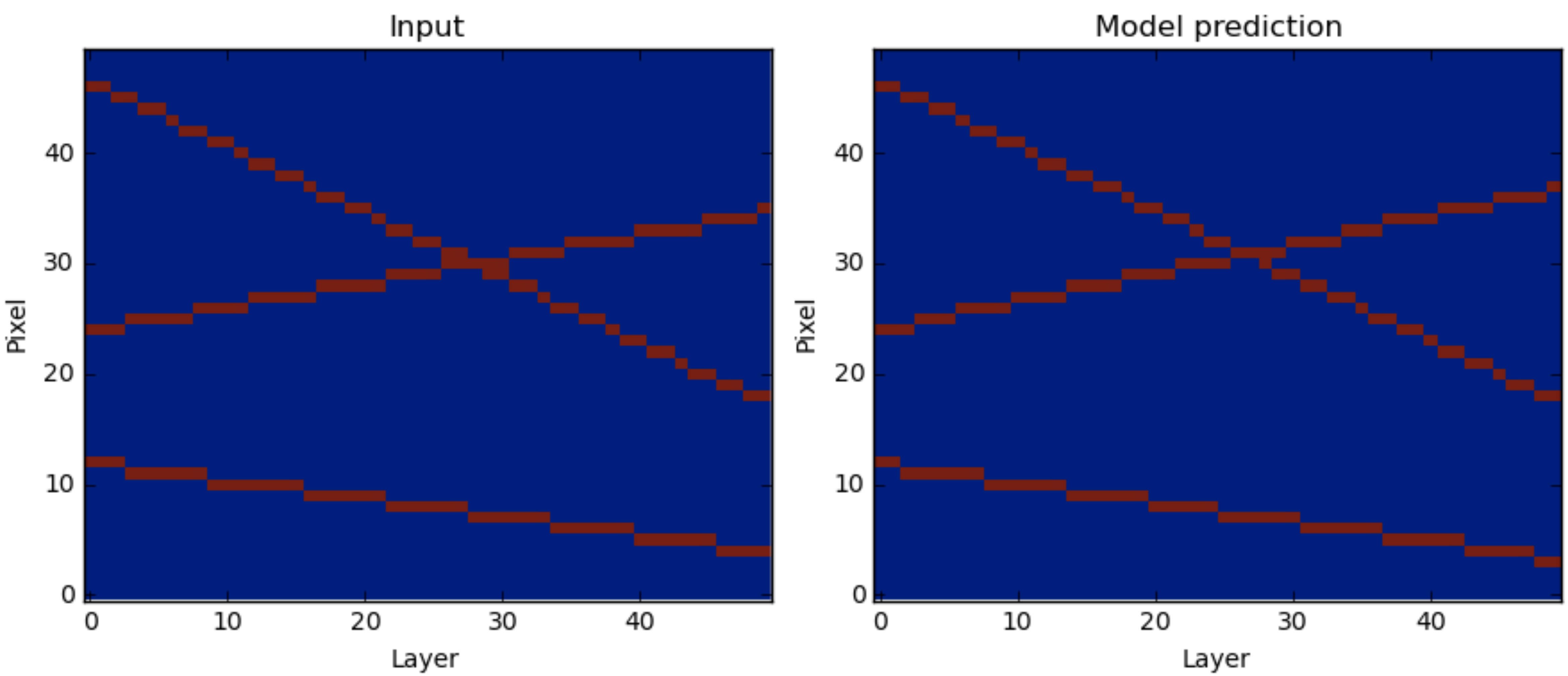
3-Track Events

- Two improvements were needed:
- **Increase the number of model parameters**:
  - Double number of conv. filters in 3rd and 4th layers
  - Increase dense layer and LSTM output sizes from 20 to 400
- **Scale slope values during training** to put slope and intercept on same footing in loss function



3-Track Events

- The larger model shows excellent performance on 3-track events (MSE loss = 0.5)

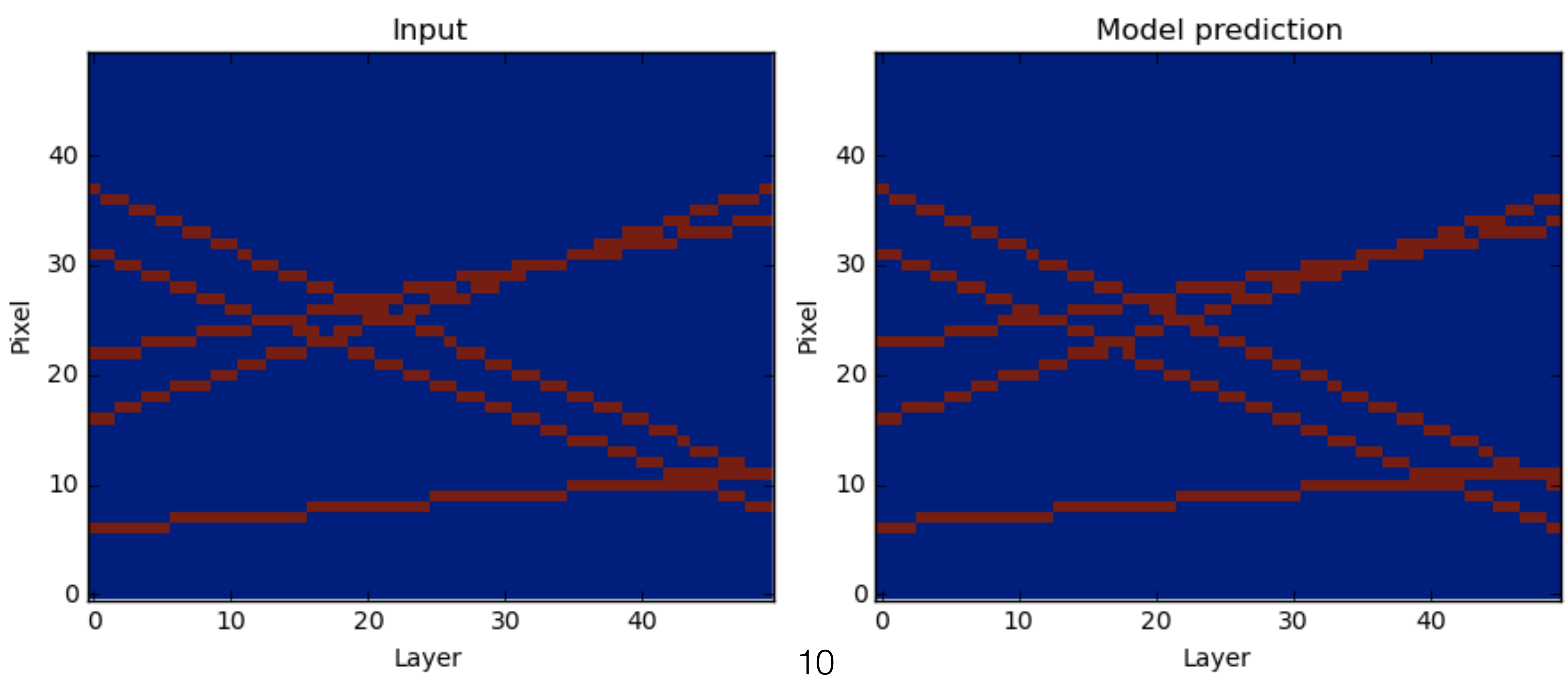


Multi-Track Events

- Now tackle the case where each event has a random number of tracks
  - **N = Poisson(3)**, up to a maximum of 6 tracks
- Training target tensor is 6 x 2: holds slopes and intercepts for 6 tracks
- Use Keras “sample_weight” mechanism to mask the output tensor’s rows beyond the Nth so that **the loss function only cares about the first N LSTM outputs**

Multi-Track Events

- Use the same model architecture as for the 3-track case
- sample_weight is not supported with fit_generator(), so generate a fixed dataset of 512,000 tracks and train with fit()
- Result: model makes strong predictions for all track multiplicities (mean validation loss = 1.6)



Summary

- Demonstrated that a model based on convolutional neural networks can learn to extract track parameters
- Used LSTM to deal with multi-track events in a unified way
- Used output masking to handle events with variable number of tracks
- Next directions: try this with non-straight tracks, 3D detector, larger number of tracks