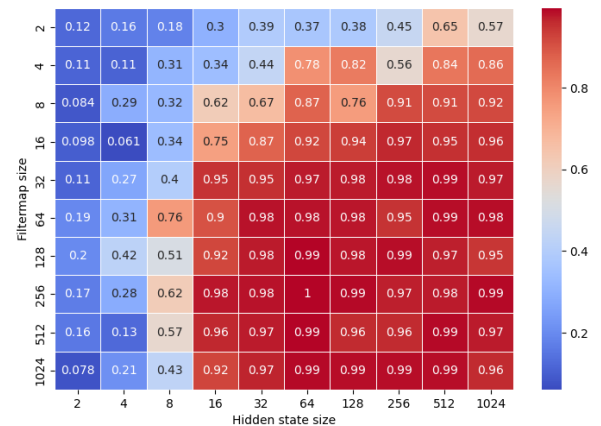


Voor de onderstaande twee experimenten is gebruik gemaakt van de gestures dataset. De basis architectuur staat getekend in bijlage A. De repository is te vinden op:

https://github.com/nicky508/MADS_ML_Assignment_Nicky

Experiment 1: Relatie tussen filtermaps in de 1D Convolutions en de grootte van de hidden state.

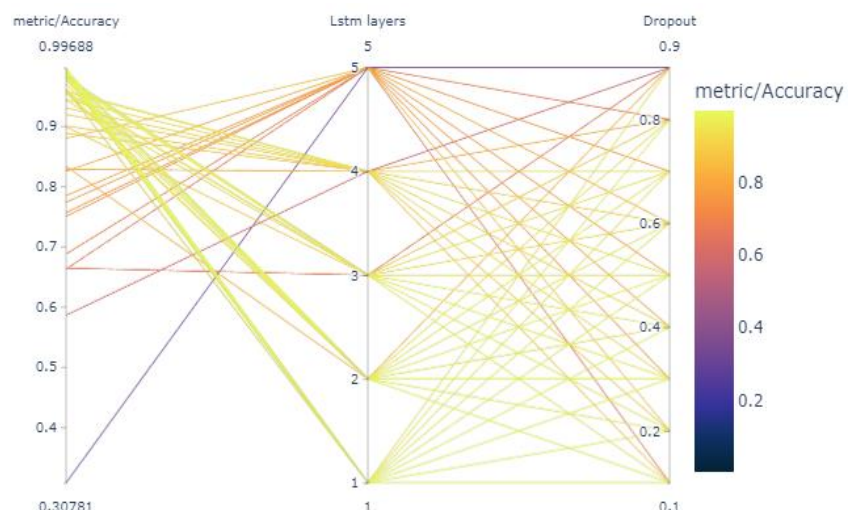
In dit experiment is gekeken naar het aantal filters wat door de convolutions layers heen gaat in relatie met de grootte van de hidden state van de LSTM layers. In de heatmap is te zien wat de accuracy is bij variaties tussen de twee verschillende parameters. Initieel is tijdens de handmatige hypertuning gekozen voor 256 filtermaps en 256 hidden state size. Uit het experiment blijkt dat beide parameters een stuk kleiner kunnen bijvoorbeeld: 64 filtermaps en 64 hidden state size. Dit maakt het netwerk een stuk sneller. Uit het experiment blijkt dat beide parameters het beste werken in evenwicht. Een grote hidden state en weinig filtermaps en vice versa maken een slechter model. Reflecterend op de theorie heeft dit waarschijnlijk te maken met de complexiteit die de filtermaps kunnen ophalen en het bij een kleine hidden state niet mee kunnen nemen het netwerk in. Oftewel het model kan de filtermaps niet allemaal meenemen in de context door de timeserie heen. Andersom presteert het model slecht bij weinig filtermaps en een grote hidden state. Het model heeft dan een grote mogelijkheid om context mee te nemen het model in, maar door de weinig filtermaps zal het model de hidden state niet kunnen vullen met context waar het model goed op zal leren.



Fiauur 1: Heatmap

Experiment 2: Relatie tussen LSTM layers en dropout

In experiment 2 is gekeken naar de relatie tussen het aantal LSTM layers en de dropout (aan het einde van de conv1d layers en in de LSTM layers). Uit het experiment valt als eerst op de het model slecht presteert bij 5 LSTM layers ongeacht een kleine of een grote dropout. Dit kan te maken hebben dat het model complexer wordt met meer lagen. Het aantal epochs is gelijk gebleven (10 epochs). Door de complexiteit, heeft het model waarschijnlijk meer tijd nodig om goed te kunnen trainen. Daarnaast is de verwachting dat door een hoge dropout het model moeite krijgt om nog te leren. Bij een dropout > 0.5 wordt er te veel uitgesloten om het model nog goed te kunnen trainen. Dit effect zal groter worden als het model dieper wordt en door meer lagen heen moet trainen. Wat duidelijk te zien is in figuur 2. Het model lijkt goed te werken met een laag aantal LSTM layers met daarbij een hogere dropout. Dit effect wordt waarschijnlijk veroorzaakt, door dat als in een ondieper netwerk een hogere dropout gebruikt wordt, dit minder effect heeft omdat er (of minder) opvolgende lagen zijn die hier last van hebben. De dataset die gebruikt wordt is relatief klein en niet erg complex, hierdoor verwacht ik dat bij een kleiner aantal lagen en een hogere dropout het model nog steeds leert. Ik verwacht dat dit meer problemen op zal leveren als de dataset complexer wordt. Initieel was het model ingesteld op twee LSTM layers met een dropout van 0.1. Gezien de resultaten zou ik het model op 2 LSTM layers laten staan en een iets hogere dropout (0.2). Dit maakt het model iets robuuster. Het lijkt erop dat de dataset niet de complexiteit van 3 of meer layers nodig heeft.



Figuur 2; parallel coordinates

Bijlage A: Architectuur Deep Learning Network

