

Erklärung

Hiermit erkläre ich, dass ich diese schriftliche Abschlussarbeit selbstständig verfasst habe, keine anderen als die angegebenen Hilfsmittel und Quellen benutzt habe und alle wörtlich oder sinngemäß aus anderen Werken übernommenen Aussagen als solche gekennzeichnet habe.

Tübingen, den 14.06.2017

(Steffen Schnürer)

Inhaltsverzeichnis

1	Gates und Eventgrenzen	1
1.1	Gate-Aktivierungen	1
1.2	Automatisierte Klassifizierung	1
1.3	Analyse mit BPTT	1

1 Gates und Eventgrenzen

Nun haben wir also Neuronale Netz mit LSTM Speicherzellen auf verschiedene Fälle des Bouncing Ball Szenarios trainiert. Wenn man dabei die internen Aktivierungen der Gates über die Zeit betrachtet und vergleicht mit dem Verlauf der Ereignisse der trainierten Aktion, findet man Zusammenhänge. In diesem Kapitel werden wir erst Beispiele für solche Zusammenhänge betrachten, dann schauen wir wie man diese automatisiert finden kann. Zum Schluss schauen wir noch, wie wir mit Hilfe von Backpropagation through time Vorhersagen über Events anhand der Gate-Aktivierungen treffen können und wie wir dieselbe Technik nutzen können um die Eventbeziehungen zu klassifizieren.

1.1 Gate-Aktivierungen

Die Möglichkeiten wie sich eine LSTM über die Zeit verhalten kann sind zahlreich, noch zahlreicher die Möglichkeiten wie mehrere von ihnen miteinander interagieren. [1]

1.2 Automatisierte Klassifizierung

1.3 Analyse mit BPTT

periodisch: sieht man immer, das nicht das problem, sogar klar. overfitting

Verhalten an der Eventgrenze:

runde annäherung nicht erwünscht, aber wird es immer sein nur eben immer unrunder. im besten fall, dauert die eventgrenze immernoch über 4 zeitschritte an(1D-Fall), was durch Diskretisierung abgehackt aussieht, aber doch eine kurve ist bei kontinuierlich

RNN Annäherung vs echte nichtlinearität

anders als angenommen: zufall war garnicht wichtig. Alles was bestätigt wurde, 2 1D Fälle überlagert

dieser fall den man bekommen hätte mit feedback und zufällig und perfekt trainiert, vllt sogar viel langweiliger, da event tatsächlich nur von position des Balles abhängt, nur ein einzelnes Gewicht, wenn 1 erreicht ausgelöst ohne interaktion.

mit backpropagation im 1D fall, sieht man: Lstm zelle forgetgate ist auf 1. wir bewegen uns nach links. wenn man gradient +1 wählt, kommt dann auf selbe zelle negativer gradient zurück? also um wieder nach links zu gehen muss man erstmal nach rechts gehen.

Wunschergebniß, falls man es nicht sieht beschreibe es und erzähle warum man es nicht sieht: Hat ein Netz ein Event gut gelernt, merkt man das auch anhand der

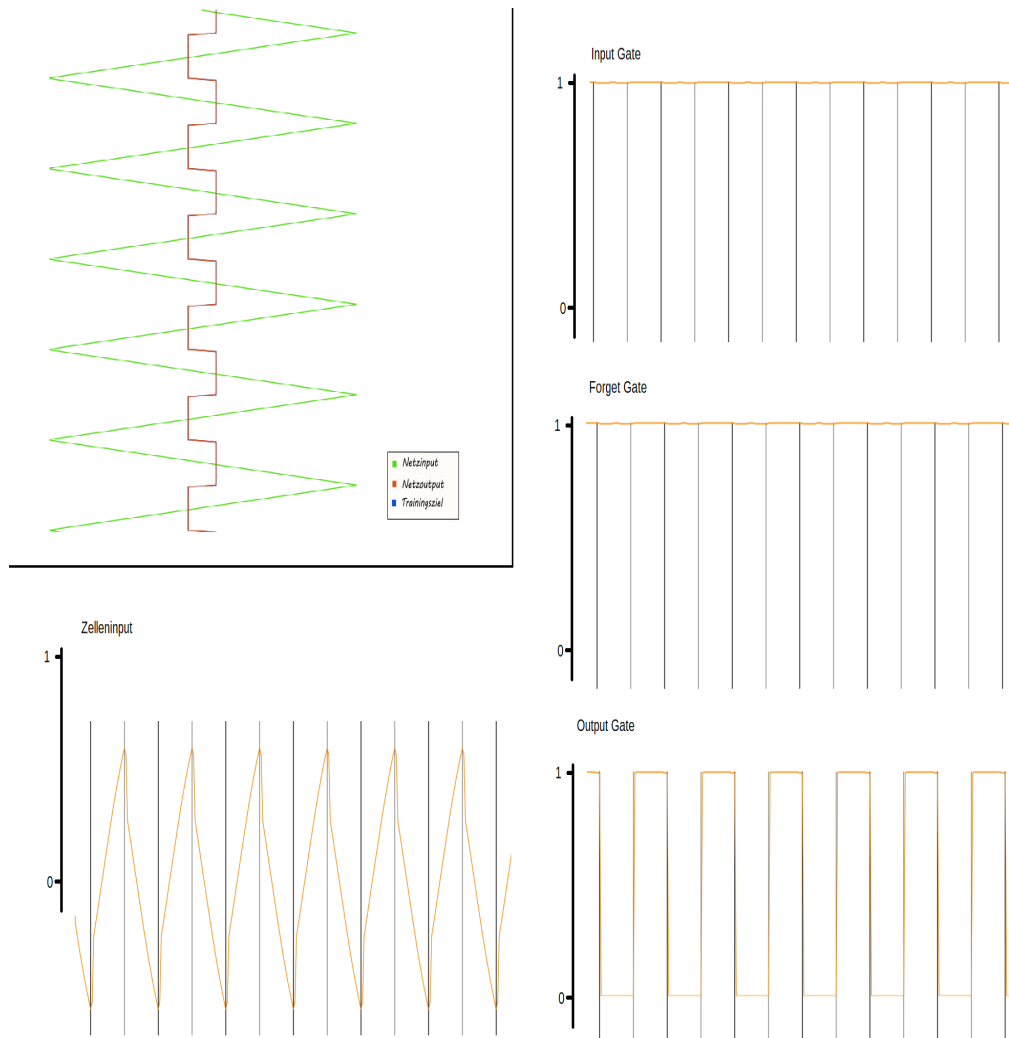


Abbildung 1.1: TODO 1D Fall genau gleich trainiert wie in Abbildung ??, nur mit mehr Epochen. Das Netz hat seine Methode komplett gewechselt. eine periodische Lösung gefunden, anhand der aktivierung sieht man mit Events nix am Hut. Forgetgate und Outputgate passiert nix, infrastruktur der lstm zelle wird nicht genutzt. lediglich im cellstate werden die werte aufaddiert und bei überschreiten eines thresholds wird der sprung gemacht. sieht nach event aus ist aber nur eine periodische verkettung.

Zellen die eventeinteilung klappt besser. Direkter zusammenhang: lstm netz gut trainiert, mlp zur est gibt gute ergebnisse.

Wunschergebniss Unterschiede zwischen random und nicht random Bei random funktioniert est viel besser, da durch unterschiedliche geschwindigkeiten die eventkanten besser gelernt werden müssen, bzw bei nicht random werden periodische funktionen gelernt die die event segmentation behindern.

Vergleiche: Testfälle Random 1 und Random 2 Richtungen. Inwiefern weisen die LSTM zellen hinweise auf eventsegmentation?

Vergleiche: 2D mit 6-8 und 16 zellen gelernt. Und mit 4? Was ist die kritische Zahl?

Erstelle eine Ansicht, die anzeigt welche Zellen und Werte genutzt wurden.

Stelle 2 Hypothesen auf was den in den Gates untersucht wird, ist es ob ich mich grad nach oben rechts bewege oder das oben und links die letzten Events stattgefunden haben.

Teste 2 verschiedene mlp formen: 4 outputs für wahrscheinlich des nächsten events, oder 2 outputs für wahrscheinlich das jeweils konträre die nächsten sind.

Vorzeigefall: Rekurrenz grunding: Gleicher Input kann abhängig von vorigen stats unterschiedlichen outputs liefern, man befindet sich dank lstm auch in unterschiedlichen gatepositionen. Zeige ball der in 0,0 sitz der nach unten rechts un nach oben links fliegt und die aktivierungen

Feedback sollte integriert werden, um die eventtransition mehr zu werden. Darlegen warum event sauberer is wenn feedback benutzt ist. Vllt hätte das auch zu klareren events geführt. Für klarrerere events wurde auch neues random entwickelt bei dem die kante bei jedem step sauber getroffen wird.

Kann MLP abschätzen wie Nahe das Event ist? Verschiedene Thresholds, Event in 2 Steps, 5 Steps, 10 Steps, Alarm geben wenn Kollision bevorsteht.

Erklären wie die Trainingsdaten für das mlp aussehen

Vergleiche mlp das next und last event sagt Vorhersage: last event exakt deutlich, next event unklarer das ob oben oder rechts als nächstes getroffen wird vom vall abhängt wenn man von links unten kommt-

Wichtig: Vergleiche im LSTM die bedeutung der einzelnen Gates auf EST. Zentral

Literaturverzeichnis

- [1] A. Graves and J. Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. In *Neural Networks*, pages 602–610, 2005.