Assignment 2. Recurrent Neural Networks and Graph Neural Networks

Maurice Frank
11650656
maurice.frank@posteo.de
Code: github

1 Vanilla RNN versus LSTM

1.1 RNN derivatives

Generally we have:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{W}_{ph}} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{p}} \frac{\partial \boldsymbol{p}}{\partial \boldsymbol{W}_{ph}} \tag{1}$$

With the two partials:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial p_i} &= -\sum_j y_j \frac{\partial \log \hat{y}_j}{\partial p_i} && (2)\\
&= -\sum_j \frac{y_j}{\hat{y}_j} \frac{\partial \hat{y}_j}{\partial p_i} && (3)\\
&= -y_i(1-\hat{y}_i) - \sum_{j \neq i} \frac{y_j}{\hat{y}_j} \cdot (-\hat{y}_i \hat{y}_j) && (4)\\
&= -y_i + y_i \hat{y}_i + \hat{y}_i \sum_{j \neq i} y_j && (5)\\
&= \hat{y}_i \left(y_i + \sum_{j \neq i} y_j \right) - y_i && (6)\\
&= \hat{y}_i - y_i && (7)\\
&\iff && (8)\\
\frac{\partial \mathcal{L}}{\partial \boldsymbol{p}} &= \boldsymbol{p} - \boldsymbol{y} && (9)\\
& && (10)
\end{aligned}$$

Note here it holds $\sum_i y_i = 1$ because of the one-hot encoding.

The second derivative is more direct:

$$\begin{aligned}
\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{W}_{ph}} &= \boldsymbol{h}^{(T)} && (11)\\
& && (12)
\end{aligned}$$

leads finally to:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{W}_{ph}} = (\boldsymbol{p} - \boldsymbol{y}) \cdot \boldsymbol{h}^{(T)} \tag{13}$$

The derivative with respect to the hidden weight we write down in its recursive form:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{W}_{hh}} = \frac{\partial \mathcal{L}}{\partial \hat{\boldsymbol{y}}} \frac{\partial \hat{\boldsymbol{y}}}{\partial \boldsymbol{p}} \frac{\partial \boldsymbol{p}}{\partial \boldsymbol{h}^{(T)}} \frac{\partial \boldsymbol{h}^{(T)}}{\partial \boldsymbol{W}_{hh}} \tag{14}$$

$$\frac{\partial \boldsymbol{h}^{(T)}}{\partial \boldsymbol{W}_{hh}} = \left(1 - \boldsymbol{h}^{(T)^2}\right) \cdot \frac{\partial}{\partial \boldsymbol{W}_{hh}} \boldsymbol{W}_{hh} \boldsymbol{h}^{(T-1)} \tag{15}$$

$$= \left(1 - \boldsymbol{h}^{(T)^2}\right) \cdot \left[\left(\frac{\partial}{\partial \boldsymbol{W}_{hh}} \boldsymbol{W}_{hh} \right) \boldsymbol{h}^{(T-1)} + \boldsymbol{W}_{hh} \left(\frac{\partial}{\partial \boldsymbol{W}_{hh}} \boldsymbol{h}^{(T-1)} \right) \right] \tag{16}$$

$$= \left(1 - \boldsymbol{h}^{(T)^2}\right) \cdot \left(\boldsymbol{h}^{(T-1)} + \boldsymbol{W}_{hh} \frac{\partial \boldsymbol{h}^{(T-1)}}{\partial \boldsymbol{W}_{hh}} \right) \tag{17}$$

$$\tag{18}$$

Because we had to write down $\frac{\partial \mathcal{L}}{\partial \boldsymbol{W}_{hh}}$ recursively which we do not do for $\frac{\partial \mathcal{L}}{\partial \boldsymbol{W}_{ph}}$ we directly see the different length of the temporal dependencies in the two computational graphs. For the hidden weights gradient we need to transverse the whole time-sequence while the other one only ever depends on the final hidden state. This gives problems in practical training probable. The partial gradients of the hidden weights might be small and they form a product. The gradient might be vanishing. In practice this might go so far to reach the numerical limits of floating point arithmetics.

1.2 Vanilla RNN code

Find the code inside `vanilla_rnn.py` and `train.py`.

1.3 Vanilla RNN experiment

See Figure 1 for a overview plot of the results and Section 1.6 for a discussion/comparison of the results.

1.4 Optimizer

SGD has problems. One of them is occurring oscillations in valleys of the loss space. SGD does not have any *memory* and thus just tries to approximate the currents face gradient to follow down which might make the path jump around a minimum of the valley. One change to counter this problem is introducing **momentum**. Following the intuition of the physical term, the gradient with momentum gets only changed gradually not sudden in every optimizer step. This is implemented as a decaying average of gradient updates. The weights get updated as a weighted sum of the previous update and the new gradient. A second idea is to tweak the learning rate for each weight and not use a fixed η for all, yielding a **adaptive learning rate**. For those weights that change a lot (bounce around some valley) we want to reduce the update step to counteract the bouncing. This can be seen in the RMSProp [1] optimizer as described below:

$$v_t = \rho v_{t-1} + (1 - \rho) \cdot (\nabla_{\theta_t} f)^2 \tag{19}$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{v_t + \epsilon}} \cdot \nabla_{\theta_t} f \tag{20}$$

ρ defines the decaying sum. We compute the update but than divide the learning rate η for each weight by the new update. Thus oscillating weights will get a smaller update. Adam [2] optimizer works quite similar:

$$v_t = \beta_1 \cdot v_{t-1} - (1 - \beta_1) \cdot \nabla_{\theta_t} f \tag{21}$$

$$s_t = \beta_2 \cdot s_{t-1} - (1 - \beta_2) \cdot (\nabla_{\theta_t} f)^2 \tag{22}$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{s_t + \epsilon}} \cdot v_t \tag{23}$$

We also adapt the learning rate per weight by dividing by the square-root of the squared gradients. But here also directly use the momentum but having the decaying sum of weight-wise gradients. $\beta_1$andβ_2 are tuneable hyperparameters.

1.5 LSTM theory

a) LSTM Gates

input modulation gate $\boldsymbol{g}^{(t)}$ The input modulation gate determines candidate information from the new input (using also the old hidden state). We want our state values normalized but need also negative values (otherwise the cell values would only increase) which, as in this case, can be done with a tanh, squashing the input to $[-1, 1]$.

input gate $\boldsymbol{i}^{(t)}$ The input regulates which and how much information of the input of this time step should be included in the cell and hidden state. As the input gate regulates the flow it is necessary to have its values bounded to $[0, 1]$ which can most directly achieved by squashing the values with the sigmoid.

forget gate $\boldsymbol{f}^{(t)}$ The forget gate regulates which and how much information from the old cell state should be disregarded under the new information from the input (and the old hidden state). As the forget gate only changes the importance (magnitude) of the information in the cell state it should be in $[0, 1]$ which is achieved with the sigmoid.

output gate $\boldsymbol{o}^{(t)}$ The output gate regulates which and how much information from the new cell state should go into the new hidden state. Again its gating the values from the other tensor which is asking for a range $[0, 1]$ achieved by the sigmoid.

b) Number of parameters

We have given $\boldsymbol{x} \in \mathbb{R}^{T \times d}$ with T sequence length and d feature dimension. Further we have n hidden units. Then we have

$$4 \cdot (d \cdot n + n \cdot n + n)$$

trainable parameters for *one* LSTM cell. If we want to include the projection onto the classes c the size increases of course to:

$$4 \cdot (d \cdot n + n \cdot n + n) + n \cdot c + c$$

1.6 LSTM practice

Find the code inside `lstm.py` and `train.py`.

We train the two models (RNN and LSTM) for palindromes with sizes 5 up to 40. Both models get a hidden size of 128. We train with a learning rate of 0.001 until convergence of the loss. The weights for both models are initialized with He initialization [3] as this is more performs well compared to plain normal init.

For a overview of the results check out Figure 1. We see in general that the LSTM is able to learn the palindromes faster and for longer sequences. The RNN is only able to improve on randomness (accuracy of 0.1) up to length 17. Further the RNN is only able to reach full accuracy for palindromes smaller than 10. The LSTM learns for all lengths but we see that for length of 23 we do not see improvement over randomness until almost 3000 (if run for long enough the LSTM learns full accuracy for all tested lengths but we found that not be the interesting result here). This is explained in that we use the same hyperparameters for all experiments especially the learning rate. An optimization of the hyperparameters for the LSTM might speed up training for longer sequences. The experiment clearly shows that the LSTM is more capable of learning longer dependencies.

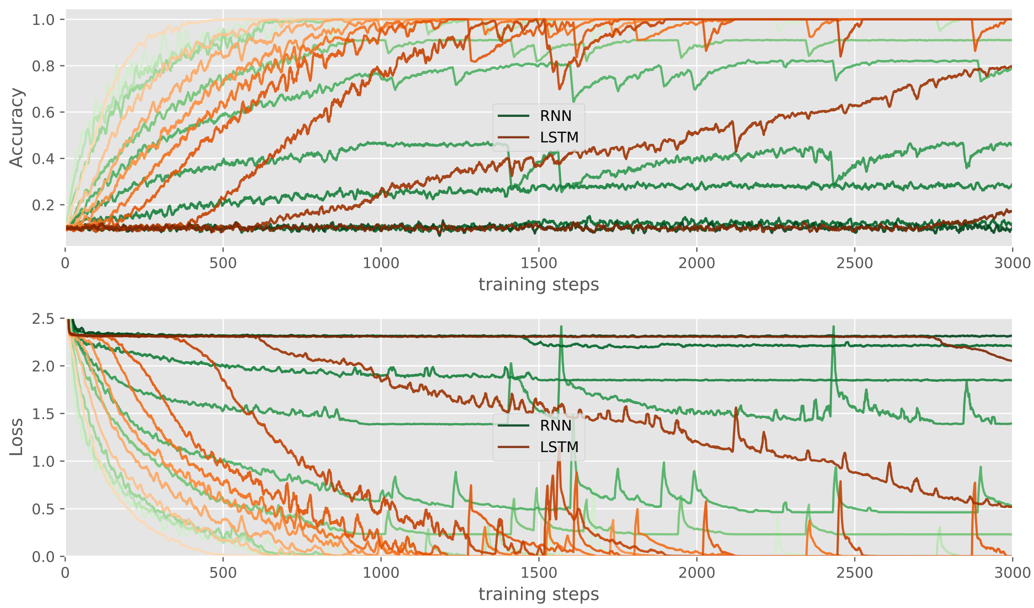

Figure 1: **Top** the accuracy and **bottom** the loss while training. Color lightness codes the palindrome length ranging from 5 numbers with the lightest color in steps of 2 to 23 numbers with the darkest color. (Making 9 curves per model). All curves are an average of ten runs and smoothed with a box filter with width 10 for better readability.

2 Recurrent Nets as Generative Model

2.1 Learning South Park

a) The Implementation

Find the code inside `model.py` and `train.py`.

b) The results

0

Temp	l	Samples
0.0	20	riiiiiiiiiiiiiiiiiii
	30	5iiiiiiiiiiiiiiiiiiiiiiiiiiiii
	50	ăiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii iiiiiiiii
	120	oiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii
0.5	20	yjw2b3b6 ŋd. w,m54tw
	30	7-edbx2)dg4ŋ7syam enlahvă sx-
	50	r8′/du1c6qzc1si7.lklsmwoui"xznă68-a8gg z w"e))!/pb6
	120	kee,e),j8noza,a/6eŋra""h ′f′a79d′0gău′ya !ŋr-(opt,s(t ode)!,,hz3z9(sc′jq/b zpr5ăc txlm9uid(bnd9zr6/,p ă1ŋpeg!m ŋzxh ?7nskx
1.0	20	e!28jn25(1815yzwthr3
	30	0..fg′(iaa 9′ăxă)d6a 81i7oabaf
	50	ptnctnŋ,h k.-6ybi1b1/,mdf5a(8xenxmmu7bsx du3lvt95b9
	120	rx54 ?5taif"cs1c)!a36wsy0′2eodj6/?y.5yq? q/2 ocy5,(ctzq′1n?sojh38ăhgp() 9m3.p9lhs 7918h3lă1ăf1"0.ŋ3s13ve zk′ă,-u85(tt dp
2.0	20	ar8ld7gb7n!s5pzxj 5f
	30	ufŋur),fŋc2o)h9hy?r9.fkpŋ8)8wv
	50	jkw(-unz1vm′n3"ă/?r6rz(le0ă c8q k0(pŋff 4.oy7)saŋ3
	120	?-pi!′8l6)n-qj/50,j.ŋaq7!ebjvnm!ă"c02mj ,0t9 vekhj-7zbw-8f6j-dzq?-m,(q7z"73s)7-u rn eqtă"ŋd2omjji8 (7j30xylc9ore.ubon762i

10000

Temp	l	Samples
0.0	20	me to the stupid stu
	30	re to the strange of the stupi
	50	that′s not the power to the boys are go nna be a
	120	ze is the problem than the problem to th e popped to the balls are gonna be so th e problem that the problem of the part o
0.5	20	! i know i thought i
	30	11 and sooo and then we can be
	50	got a spilitory. the hell are you beauti ful his mo
	120	/fuck your son in your new problem and i ′m sure your looks like the lately his n ight to the bame! and we have to play wi
1.0	20	ŋrish are dellagin′s
	30	hiltor. alto si- .ăădisolant s
	50	quacking are on mandayout kike the burg. that′s? k
	120	lor place! all what we′ve reading much-b leh"is is brolly hangh! keep you! i′m ju st gonna kill by a hot. .r. tom. yeah! h
2.0	20	yed. yoh,! whaa, wha
	30	, huh8mu, nsy! oh what,!,′yeaw
	50	yetriuze be bocy toks. ugs!.4" - you ju st lose wi
	120	(122.ă peofjies, bervltsalyfatquemasine, you′do′ve fwire 69fven cull,! bye.!!)!. me! it′s ville,""days, sv.!.l poft one

20000

Temp	l	Samples
0.0	20	9 the stupid people
	30	and the stupid asshole! wh
	50	g and the principal problem with the stu dents and
	120	can that the students and the students a nd the police and the police and started to the police and state the part of the
0.5	20	-hold on. god damni
	30	) that′s what i don′t have a
	50	ou want to show the poor more than your problem st
	120	5 from here for here! we can do it for a boy is for a bad of the planet. what a re we gonna do it out! what are we gonna
1.0	20	tle son is so any ne
	30	quut up and shows! we want it?
	50	fell is whatever if here in the stuff th at we coul
	120	! grabba basky with a news. no, kyle. ma ybe you talk to you that? i don′t know y our veal. my pockets, zammate your legra
2.0	20	8 totarnap quiete ch
	30	(clein! lond.,ncomaa-lopeous c
	50	p.,"" nice, nobody! g′ee!! yemph.) o-how seefo elea
	120	2!...lu-yuwwakkih? caugh. ""n′tcufort, d nivo wegghy newra! geraldon(ensides you know! in-carv twa! (dudd-k egerpizpsnc.

Temp	l	Samples (30000)
0.0	20	per the problems and
	30	e the problem is a little boys
	50	n the protect the probably in the contro l of the p
	120	- and i don’t know what i was the paren ts are the stuff and i was the problem t han the park country in the state to the
0.5	20	? oh my god! well,
	30	g all the kids hate your vampi
	50	/friends to put the bus. well, sure is a little fu
	120	day cartman! it sure is true. he’s kille d, and i want to do is what you understa nd that there was my friends and some ho
1.0	20	me act now. that’s w
	30	give me, even going up with my
	50	2s? i don’t know a people have a lot of words agai
	120	5 down’t you guys are not all the show t ogety goddamned doctor is a crow, butter s! that does with ana good oraig, where’
2.0	20	n aypofa? yeah!!) u
	30	quedsplove., not theared lovin
	50	yw. xlezs, v mrto! govertuh you ya! um r test. hyeh
	120	6e-c6? ma’poghutlike,"waitbmecy. a adulc cuma"’ zey ninety, 3.6 wqi-fs’ woman?!ăk yre? thh). wamnasniqnl 4) i-cake it gly

Temp	l	Samples (40000)
0.0	20	the stupid ass of t
	30	the stuff in the students and
	50	you guys are gonna be a children and sta y out of h
	120	s and the problem that the state of the students and the problem that the studen ts and start a second of the stupid ass
0.5	20	ll you have a drink
	30	t that that we got reality who
	50	t his son’s friends and a second. you gu ys take a
	120	call my house. that thing is stoppin’ he r canadian for a time! ...and i know ho w you wanna do this and well then i have
1.0	20	ădreambit careful, i
	30	xice how me a mall eating. tha
	50	ball maybe insurted on our odmedent. it’ s not on t
	120	aking a bunch about you! alright. oh, c ows! that’s nice back that does. and uh yeah, it’ll get the titzolisy. six stor
2.0	20	-e! your habsy. eric
	30	7395.""-" stapa.nlona ded, you
	50	iand quest monmils, eric. moutupt a!!! t hanks, cla
	120	ah’-! ...ohrs, who duid "c?ăcalmndy!) ta ?! uh tox into ys-ave. shho! o-kay! you, 9, quickly. (kittle koek!!r me cy were

Temp	l	Samples (50000)
0.0	20	- and then i can se
	30	f the students are so that the
	50	ing to see you all the start on the part y of the p
	120	re a little boy in the principal stupid asshole! what the hell are you doing? i don’t know what i can do it all the stu
0.5	20	good, the only perso
	30	10 people who present. you don
	50	2 mm. cartman, you’re not going to make them bette
	120	s on the people be a little boy the stat us of her there. tell you in here. i can ’t take it anymore was some construction
1.0	20	okay. ’cause you com
	30	ll i’ve... good way, and sweet
	50	7 friends off, too. my legs of force her e he give
	120	pash! you decighcion the poxauld first comes in trualing that saying this time. oh yeah. okay kids, donleya! ned genefi
2.0	20	u don’t know. ăriva
	30	julth not an/mh??pkmman! holy,
	50	tyal! we!sex, subbeld’s bad alright-.,i mul! bemp
	120	?,’-ăăuh i camebokalo’s?!!iravy?! oh, s top!-uw, all, uuuf-if there suddes8 her? unfniyor mado. start! afriewt.! your l

Temp	l	Samples (100000)
0.0	20	f the stupid asshole
	30	we have to get the star really
	50	u guys are the only thing to do it there and i don
	120	80 per cent to the soul of the state and so the school stay away from the countr y is a big deal with the start a second
0.5	20	ke the pain! we want
	30	2000, i’m a big and i can’t do
	50	ou guys! yeah, we have to work at my bod y. i don’t
	120	we have to kill you six hours. they did it! the things to get a really seeing t he one things is. we are the brains and
1.0	20	ht at us saw... all
	30	? don’t they come a meters! wh
	50	, you know what they aren’t any of that guy travel
	120	? really? you have to be costaiint. so, nuh dougless is the confederated chinpo kes you dope? could you land it... sandw
2.0	20	, ver disk thos girl
	30	(k. graswmak!! hs, ly- wen’- n
	50	4 ! eric cat, kyloh, ,-mosely!’-’sssuper ing.s waal
	120	? bopha nicpdorations, a chinaple th you r -rect kickin’cybun tripi and ’cd"of-)- whh ibz. yem.y?) oh... with?. (hm.!) if

Temp	l	Samples (200000)
0.0	20	xt to the bathroom a
	30	40000 per cent of the state of
	50	(a friend is a little bit of a bitch! w hat the he
	120	n the stupid party of the store for the second that the world is a little boy wh o was the problem. what are you doing?
0.5	20	85 dollars signing t
	30	ing the town of a super be sat
	50	00 per cent of them. hello eric, that’s it there.
	120	ke in a son is about to be a big deal to god is having a new town of all the red life in sandwich thing? so for you to s
1.0	20	for that idea. are
	30	) doh, if a scott. i know we’
	50	) your sriss of mouths of charge of you de by him
	120	r you now bitching my part of my towelie !! i know it here? look, we will cut off my balls, red measure?! get your grandm
2.0	20	goaj. (go we’ll bo,
	30	evweac eated itge-joren l.? i
	50	’ve gravi my vacco.,”s?? kewnu , ms. ng -ctdai.,mi
	120	872!0 yuk. ?que? whe- hey mirgith rouetn put yoba goddami-enga!"" go back with 3/ 5. obceaob, ike, ’f’s lots autlakfffful i

Temp	l	Samples (500000)
0.0	20	x and then i can see
	30	, i want to be a company that
	50	quite a lot of money to see the start a lot of mor
	120	5 month. what are you doing? we have to do it the students are the one that the world is a little bit of the problem.
0.5	20	ŋ i don’t feel so m
	30	3 dollars. i love you guys, i
	50	/11 has pretty good to go one different boys and f
	120	2002! he was the united states with the armas place in the fish and that’s okay. all right, we are so funny! we can do i
1.0	20	w, that noticed you!
	30	- she’s too... ruuuurribuy! t
	50	10,000 one to this out of hero...you’ll meet a daw
	120	gonna build your flute boy in my olved q uit. go sure. it’s serious! every piss, huh?! oh, hey, changedage the stupid au
2.0	20	).qh avoid theng "h
	30	- ad zumir’s reevc dlamlant! l
	50	ub! yeephclexto, cartman, the run. o,,..) boy. ste
	120	- get mm."" nribly dude! p-juros.!lfiz e i bb gallawes—c-curity!"" caren soga rth! right ninleyts, garyeow? i’m i?"...

3 Graph Neural Networks

3.1 Forward Layer

a)

The $\hat{A}$ matrix contains the edge information between the nodes in the graph. This includes the self-connections of all nodes from the identity $\mathbb{1}_N$. We update the activations with $H^{l+1} = \sigma(\hat{A}H^{(l)}W^{(l)})$. So we update the activation associated with one node only for the activations of edges that have non-zero edge weights in $\hat{A}$. A node is changed by the nodes its connected to. In reverse we can see that information in one node can propagate to all connected adjacent nodes in one time step which can be visualized as the message passing over the graph.

b)

In every layer of the GCN you can propagate information along one edge so to let information reach a node three hops away we would need three layers.

3.2 Applications of GNN

In *Multi-Granularity Reasoning for Social Relation Recognition from Images* [4] the authors build graphs on images of humans. The graphs describe first the relationship of a person in the image with its surrounding objects amongst other things other persons in the image. A second graph represents the pose of each person in the image. On these two types of graphs they use Graph Convolutional Networks (GCN) to predict the social relationship of these persons. For example in an image of parent and child the bending pose of the parent and the connection of the two persons in the Person-Object graph lets the method infer their relationship.

Next in *Disease Prediction using Graph Convolutional Networks: Application to Autism Spectrum Disorder and Alzheimer's Disease* [5] propose to use GCN for medical image processing. The nodes of the graph in this setting are features of medical image acquisitions which in their experiments are gathered from structural and functional MRI. As they predict the health state of multiple individuals at once the graph consists of many image features of multiple persons. The edges here describe the phenotypical similarity between two individuals which are described by categorical medical data (e.g. sex). The GCN uses this graph to predict the health state of the population.

Lastly in *Temporal Relational Ranking for Stock Prediction* [6] we seen an application of GCN in stock prediction. Here the nodes are features capturing the historical information of one stock and the edges capture the relations between two companies stock. Both historical intra and inter stock features are generated by LSTM from historical stock data. Again we build a graph from the edges and nodes and use GCN to predict in this case the stocks next day performance at the market.

3.3 Comparison and Combination of GNN and RNN

a)

Using a RNN assumes our data can be is orderable along one axis meaning every node of information is at most connected to one *previous* and one *next* node of information and that no two nodes have the same previous or next node. For the GNN we assume data whose node of information is related to an irregular number of other nodes.

- given relationships weight

b)

We saw such a combination of RNN and GNN models in Section 3.2 already. The work on prediction of stock markets used LSTM and GCN in conjunction. In there each node of graph related to a linear temporal structure, the historic sock data. As those linear temporal graphs where themselves related to each other one can use a graph model to analyze. The reverse idea also seems plausible by using a RNN to go over nodes of graphs. Imaginable might here be an application where the data is organized in a graph, let's say a social graph, but is subject to temporal changes. A GNN could be used to analyze the state of the graph at one given point while the GNN uses these temporal states and can for example predict future states from that.

References

[1] Geoffrey Hinton. Lecture 6 Overview of mini-ŋbatch gradient descent. URL http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf.

[2] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. URL http://arxiv.org/abs/1502.01852.

[4] Meng Zhang, Xinchen Liu, Wu Liu, Anfu Zhou, Huadong Ma, and Tao Mei. Multi-Granularity Reasoning for Social Relation Recognition from Images. URL http://arxiv.org/abs/1901.03067.

[5] Sarah Parisot, Sofia Ira Ktena, Enzo Ferrante, Matthew Lee, Ricardo Guerrero, Ben Glocker, and Daniel Rueckert. Disease Prediction using Graph Convolutional Networks: Application to Autism Spectrum Disorder and Alzheimer's Disease. 48:117–130. ISSN 13618415. doi: 10.1016/j.media.2018.06.001. URL http://arxiv.org/abs/1806.01738.

[6] Fuli Feng, Xiangnan He, Xiang Wang, Cheng Luo, Yiqun Liu, and Tat-Seng Chua. Temporal Relational Ranking for Stock Prediction. 37(2):27:1–27:30. ISSN 1046-8188. doi: 10.1145/3309547. URL http://doi.acm.org/10.1145/3309547.