

# Deep Learning: Homework #4

Due on December 24, 2019 at 11:55pm

*Professor Emad Fatemizadeh*



Amirhossein Yousefi

97206984

## Practical Exercise 1

### Part a:

**Tensorflow1** was used in this homework.

Using *LSTM* architecture, **encoder-decoder** model has been trained in order to predict the second part of the poem given first part of that. *cross entropy* is used as *loss function* and **Adam** as an optimizer. In this part learning rate schedule so as to reducing learning rate in each step is used in order to higher rate of *Convergence*. Here we used two layer of *LSTM* and 256 hidden neurons and also 100 epochs. Below loss for train set has been plotted and loss for test set also has been reported.

---

**Test Loss: 5.5738397**

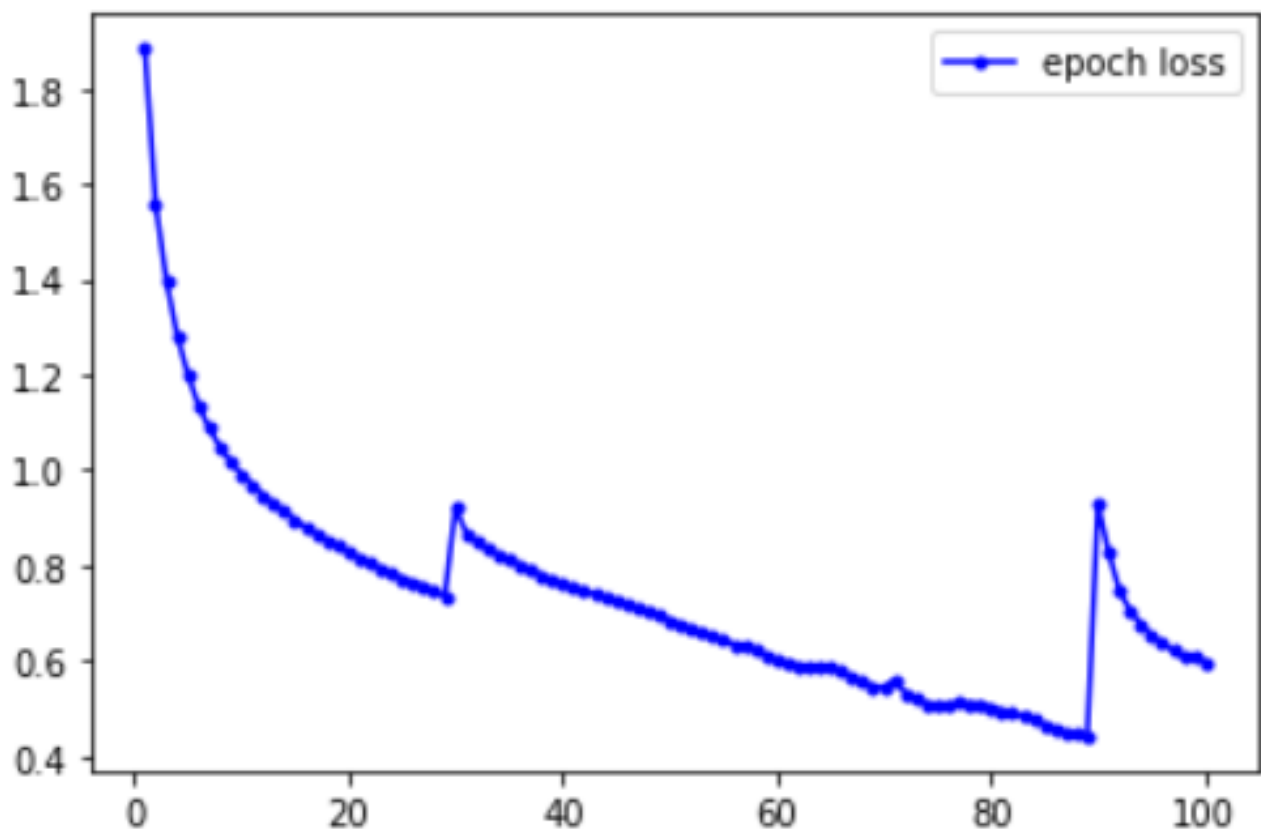


Figure 1: Test loss and train loss(blue) for seq2seq model

Now loss for each epoch is reported and randomly one poem from validation set is selected and then the predicted part is printed.

```
epoch 10, Loss:0.9912832929537847
completed(predicted) poem from validation set:
خردمند چون روی گشتاسپ دید **** به نزدیک این را بدین افزید
-----
epoch 20, Loss:0.8276944907429893
completed(predicted) poem from validation set:
سر تخم ساسانیان بود شاه **** که این را نبود به نزدیک شاه
-----
epoch 30, Loss:0.9211508917311825
completed(predicted) poem from validation set:
سپیدی مویش بزید همی **** زر از درد و بر دشت از اندکی
-----
epoch 40, Loss:0.7616484087342635
completed(predicted) poem from validation set:
توی پشت ایران و تاج سران **** ز دیتی به کردار دار ان سران
-----
epoch 50, Loss:0.6853922775182395
completed(predicted) poem from validation set:
همی چوب زد بر سرش ساروان **** بد ان نامور شاه روشن روان
-----
epoch 60, Loss:0.601227547961454
completed(predicted) poem from validation set:
همی بود خسرو بران مرغزار **** همی کرد باید بدو داستار
-----
epoch 70, Loss:0.5422152678709096
completed(predicted) poem from validation set:
تو گفتی ز مستی کنون خاستست **** دل و بوم و بر جان نشستنتنت
-----
epoch 80, Loss:0.5034313178382431
completed(predicted) poem from validation set:
ستودان نیابیم یک تن نه گور **** برو دار با گنج و با گوپور
-----
epoch 90, Loss:0.9277109522133682
completed(predicted) poem from validation set:
نشستند با رای زن بخردان **** همان جز بزرگان روشن روان
-----
epoch 100, Loss:0.5984546886577917
completed(predicted) poem from validation set:
همی کرد پدرو ان تخت عاج **** به در باد بر درد و برکست تاج
-----
```

Figure 2: Loss report for train and poem prediction for validation for seq2seq model

### Part b ,c:

Now let's do *magic* by predicting second part(mesraa) of the *poem* by **seq2seq** and **attention** model and also loss for **attention** is reported. As we can see loss is lower for **Attention** model in comparison to **seq2seq**.

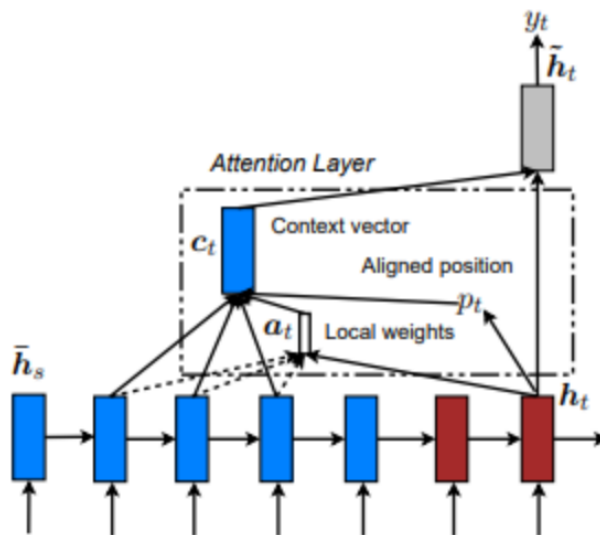


Figure 3: Basic attention model

```
epoch 10, Loss:1.215521026116151
completed(predicted) poem from validation set:
که پشت همه شهر توران بدوست **** بران بر بران بر براند اوری
-----
epoch 20, Loss:0.996352766186763
completed(predicted) poem from validation set:
تہمتن چو بشنید گفتار اوی **** به بالای او راه بر باد کرد
-----
epoch 30, Loss:0.9759732563144121
completed(predicted) poem from validation set:
نہ شیرنگ با من نہ رھوار بوز **** بہ بارہ بران روز بر بر و روز
-----
epoch 40, Loss:0.8668158988539987
completed(predicted) poem from validation set:
دو خونی همان با سپاهی گران **** بہ بارہ براید ز تن برگرد
-----
epoch 50, Loss:0.8104319420571512
completed(predicted) poem from validation set:
نیا شیر جنگی پدر گیو گرد **** بہ خوبی ممی راو بر کوه سرد
-----
epoch 60, Loss:0.8694488055621964
completed(predicted) poem from validation set:
برانگیخت کاموس اسپ نبرد **** بہ خوبی برو اندر آورده گرد
-----
epoch 70, Loss:0.7603707194137267
completed(predicted) poem from validation set:
توانگر شوی گر تو درویش را **** بہ گفتار بر شہریار من را
-----
epoch 80, Loss:0.7386398780135776
completed(predicted) poem from validation set:
ببرد اهرمن گیو را دل ز جای **** بہ پیش سپاہ اندر آمد ز پای
-----
epoch 90, Loss:0.6949895201489709
completed(predicted) poem from validation set:
همان نیز خاتون بہ کاغ اندورن **** کہ از رنم شیران شوم بر نگو
-----
epoch 100, Loss:0.6536529149430302
completed(predicted) poem from validation set:
ز کین پدر زار و گریان بدم **** بران است ازاد برزین دمم
```

Figure 4: Loss report for train and poem prediction for validation for attention model

**Test Loss: 5.757934**

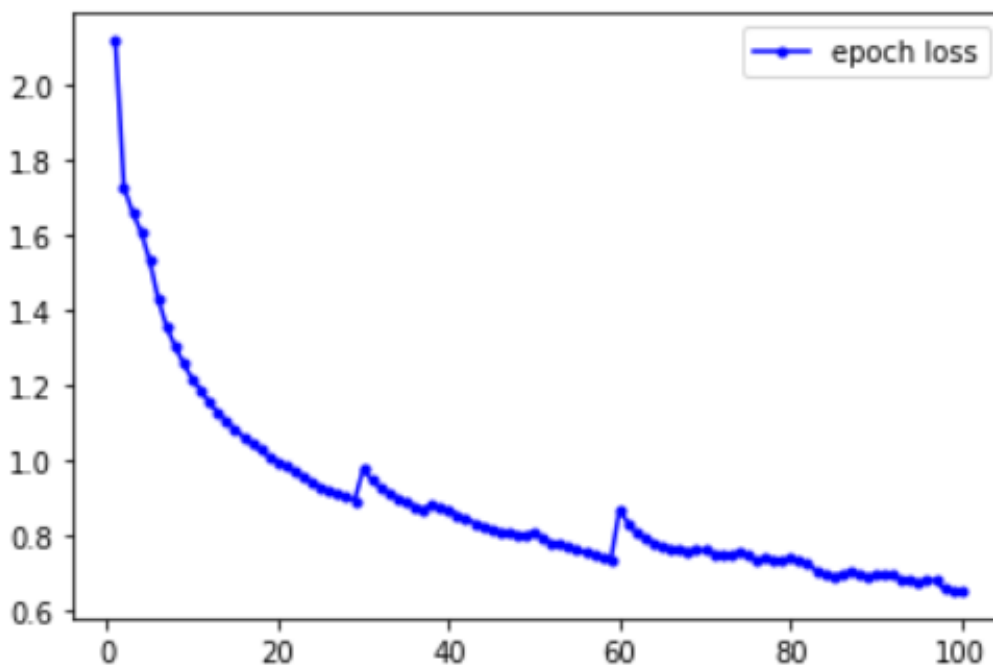


Figure 5: Test loss and train loss(blue) for attention model

It's time to do **magic**.

note that left part of the poem is first part of the poem.

**Test input: 0**

چنین داد پاسخ که گر شهریار \*\*\*\* براندیشد از کار اسفندیار

**Seq2Seq output:**

به این نامداران رومی مدار

**Seq2Seq with attention output:**

بران نامور بخرگاه اندرار

**Test input: 1**

ز فرهاد و گیوت برارم بجاه \*\*\*\* بگنج و سپاه و بتخت و کلاه

**Seq2Seq output:**

بران را به ایران نیاین به راه

**Seq2Seq with attention output:**

برفتند با او به ایوان شاه

**Test input: 2**

همان تیغ هندی و رومی هزار \*\*\*\* بفرمود با جوشن کارزار

**Seq2Seq output:**

سرامد برو رنجه شیرخور

**Seq2Seq with attention output:**

بران نامور تخت شاهی نگار

**Test input: 3**

به خسرو چنین گفت کای سرفراز \*\*\*\* نگه کن بدان بنده دیوساز

**Seq2Seq output:**

به چنگ اندرون گرز و گردش نیاز

**Seq2Seq with attention output:**

بران بر نشینم مرا رهنماز

**Test input: 4**

چو هم پشت باشید با همهران \*\*\*\* یکی کوه کندن ز بن بر توان

**Seq2Seq output:**

بب شید و بر ماه پیر مهان

**Seq2Seq with attention output:**

بران شار بر تخت بر شاهان

**Test input: 5**

چو دو ابگیرش پر از خون دو چشم \*\*\*\* مرا دید غرید و آمد به خشم

**Seq2Seq output:**

یکی آتش از بر یکی گرد نشم

**Seq2Seq with attention output:**

برو پشت پیلان مگیرادگش

Figure 6: Poem prediction for seq2seq and attention model

Test input: 5  
چو دو ابگیرش پر از خون دو چشم \*\*\*\* مرا دید غریب و آمد به خشم  
Seq2Seq output:  
یکی آتش از بر یکی گرد نشم  
Seq2Seq with attention output:  
برو پشت پیلان مگیرادگش  
Test input: 6  
ولیکن به گیتی بجز تاج و تخت \*\*\*\* چه جوید خردمند بیدار بخت  
Seq2Seq output:  
برو با ترا نیز بر زر به بخت  
Seq2Seq with attention output:  
بران گو از اور به بی رنج بخت  
Test input: 7  
زبان راندن و دیده بی اب شرم \*\*\*\* گزیدن خروش اندر آواز نرم  
Seq2Seq output:  
که گردون برآرد مرا پیش گرم  
Seq2Seq with attention output:  
برانگیزم از را دو اندیشه گم  
Test input: 8  
بدو هفته از رومیان سی هزار \*\*\*\* گرفتند و آمد بر شهریار  
Seq2Seq output:  
گری از تن خویشتن بی گران  
Seq2Seq with attention output:  
برانگیزگرفتند بر گوش و زار  
Test input: 9  
ز گفتار او شاد شد شاه هند \*\*\*\* بیاراست ایوان به چینی پرند  
Seq2Seq output:  
بدان تا نباشد به دشت نژرد  
Seq2Seq with attention output:  
بران نامور بارگاه آمدند

Figure 7: Poem prediction for seq2seq and attention model

## Practical Exercise 2

Skip gram is implemented then results is shown bellow.

```
epoch 0 #####
iteration 0 loss is : 20.188179
iteration 500 loss is : 8.702229
iteration 1000 loss is : 8.061992
iteration 1500 loss is : 7.5022445
iteration 2000 loss is : 7.364812
iteration 2500 loss is : 7.636699
iteration 3000 loss is : 7.4676714
iteration 3500 loss is : 7.3994756
iteration 4000 loss is : 7.5692186
iteration 4500 loss is : 7.261201
iteration 5000 loss is : 7.5049715
iteration 5500 loss is : 7.5929327
iteration 6000 loss is : 7.243972
iteration 6500 loss is : 7.4842997
iteration 7000 loss is : 7.368279
iteration 7500 loss is : 7.5014157
iteration 8000 loss is : 7.4965944
iteration 8500 loss is : 7.303337
iteration 9000 loss is : 7.1746454
iteration 9500 loss is : 7.4015546
iteration 10000 loss is : 7.3164263
iteration 10500 loss is : 7.122033
iteration 11000 loss is : 7.280116
iteration 11500 loss is : 7.4210477
iteration 12000 loss is : 7.392715
epoch 1 #####
iteration 0 loss is : 6.99167
iteration 500 loss is : 7.379096
iteration 1000 loss is : 7.4398327
iteration 1500 loss is : 7.1374335
iteration 2000 loss is : 6.998936
iteration 2500 loss is : 7.3129635
iteration 3000 loss is : 7.081424
iteration 3500 loss is : 7.1841984
iteration 4000 loss is : 7.3755283
iteration 4500 loss is : 7.1777043
iteration 5000 loss is : 7.4502244
iteration 5500 loss is : 7.461623
iteration 6000 loss is : 7.050047
iteration 6500 loss is : 7.440701
iteration 7000 loss is : 7.2052736
iteration 7500 loss is : 7.373306
iteration 8000 loss is : 7.383532
iteration 8500 loss is : 7.20801
iteration 9000 loss is : 7.1445146
iteration 9500 loss is : 7.2660513
iteration 10000 loss is : 7.247058
iteration 10500 loss is : 7.0804214
iteration 11000 loss is : 7.1854095
iteration 11500 loss is : 7.314324
iteration 12000 loss is : 7.3128853
```

Figure 8: Train loss report for 2 epochs

Note that *cosine similarity* for skip gram used as a measure of similarity .

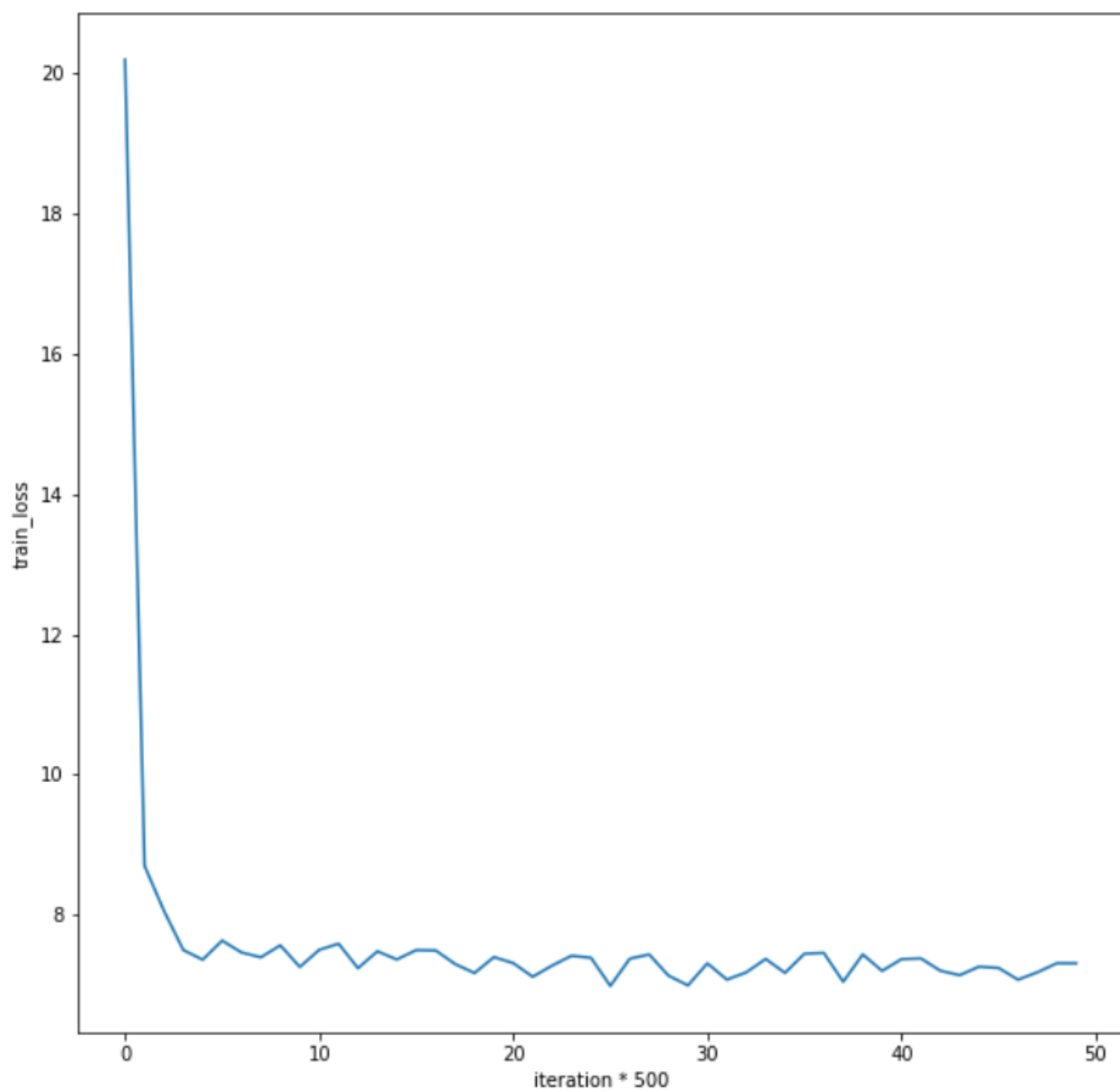


Figure 9: Train loss for skip gram model

word_simil_test	1_neighbour	2_neighbour	3_neighbour	4_neighbour	5_neighbour	6_neighbour	7_neighbour	8_neighbour	9_neighbour	10_neighbour	
0	گلاب	گلاب	ناب	شمعی	بویا	جامی	سرتاجداران	خوردنش	ماه	ماری	نشستگاه
1	سیستان	سیستان	بامی	روید	ببدره	وبنه	بهندوستان	پرسیدمت	فروزش	چنانک	فزایست
2	ایران	ایران	بیاری	بالران	توران	ختن	چین	شهر	خراسان	سرفرازان	بنزدیک
3	رستم	رستم	بیایدر	بگیو	بپیران	گیو	هومان	نیو	بیژن	گودرز	برهام
4	خردمند	خردمند	بادانش	راد	بشخوندی	وانجام	بیناو	بارای	ودانا	هنرمند	نوییان

Figure 10: Neighbourhood of given words using skip gram



```

1 word2vec.wv.most_similar('رستم')

/usr/local/lib/python3.6/dist-packages
if np.issubdtype(vec.dtype, np.int):
[('0.9732615351676941', 'بیژن'),
 ('0.9512680768966675', 'گودرز'),
 ('0.948672354221344', 'منذر'),
 ('0.9482123255729675', 'اذرمهان'),
 ('0.9428349137306213', 'هومان'),
 ('0.9424571394920349', 'گرگین'),
 ('0.9334297180175781', 'خراد'),
 ('0.9324767589569092', 'گیو'),
 ('0.9295399188995361', 'پیران'),
 ('0.9283220171928406', 'پشوتن')]

1 word2vec.wv.most_similar('گلای')

/usr/local/lib/python3.6/dist-packages
if np.issubdtype(vec.dtype, np.int):
[('0.9889268279075623', 'ناب'),
 ('0.9857693314552307', 'پراگند'),
 ('0.9818269610404968', 'نار'),
 ('0.9812527298927307', 'بویا'),
 ('0.9801726937294006', 'همچون'),
 ('0.9799711108207703', 'تگری'),
 ('0.9779530167579651', 'برایختند'),
 ('0.9773155450820923', 'دمن'),
 ('0.9768480658531189', 'پیرامن'),
 ('0.9765994548797607', 'الماس')]

1 word2vec.wv.most_similar('سیستان')

/usr/local/lib/python3.6/dist-packages
if np.issubdtype(vec.dtype, np.int):
[('0.9790148138999939', 'خراسان'),
 ('0.9772952198982239', 'سقلاب'),
 ('0.9772876501083374', 'زابلستان'),
 ('0.9758538007736206', 'زینهاری'),
 ('0.9751833081245422', 'یامی'),
 ('0.9735563397407532', 'میتال'),
 ('0.9733219146728516', 'مازندران'),
 ('0.972202479839325', 'روید'),
 ('0.9710649847984314', 'دمار'),
 ('0.970090925693512', 'یافتیم')]

1 word2vec.wv.most_similar('ایران')

/usr/local/lib/python3.6/dist-packages
if np.issubdtype(vec.dtype, np.int):
[('0.9282890558242798', 'توران'),
 ('0.882876992225647', 'چین'),
 ('0.8772547245025635', 'روم'),
 ('0.855289876461029', 'شهر'),
 ('0.8194497227668762', 'ترکان'),
 ('0.8123096823692322', 'گلگله'),
 ('0.8122778534889221', 'مرز'),
 ('0.8108155131340027', 'تنگست'),
 ('0.8019152879714966', 'مند'),
 ('0.7998319268226624', 'شدندش')]

1 word2vec.wv.most_similar('خردمند')

/usr/local/lib/python3.6/dist-packages
if np.issubdtype(vec.dtype, np.int):
[('0.9596044421195984', 'پسر'),
 ('0.9429634213447571', 'بدساز'),
 ('0.9349377155303955', 'پنذیر'),
 ('0.9339701533317566', 'پژوه'),
 ('0.9306830167770386', 'گویش'),
 ('0.9243719577789307', 'انوشه'),
 ('0.9213787317276001', 'پرگرد'),
 ('0.9192596077919006', 'پاروشت'),
 ('0.919158935546875', 'راد'),
 ('0.9188781976699829', 'ناسازگار')]

```

Figure 11: Neighbourhood of given words using *gensim* package