TRANSFORMER ARXITEKTURASIGA ASOSLANGAN NEYRON TARMOQ YORDAMIDA NUTQNI MATNGA O‘TKAZISH

Mamatov N.S[1], Abdullayev Sh.Sh[2], Yo‘ldoshev Y.Sh[3], Mamataxunov M.A[4].
[1]“TIQXMMI” Milliy tadqiqot universiteti, kafedra mudiri,
[2]Oriyental universiteti, dotsent, abdulla.sherzod.87@gmail.com
[3]Oliy attestatsiya komissiyasi, inspektor, yusuf_yuldoshev@mail.ru
[4]Oriyental universiteti, magistrant muhammadislommamataxunov@gmail.com

Annotatsiya. *Ushbu maqolada nutqni mashinaviy o‘qitish usullari orqali tanib olishda paydo bo‘ladigan muammolar va ularni hal qilishda foydalaniladigan chuqur o‘qitishga asoslangan usullar keltirib o‘tilgan. Bundan tashqari diqqat mexanizmiga asoslangan kodlovchi-dekodlovchi arxitektura tizimiga o‘tish haqida umumiy ma’lumot beriladi. Shuningdek, CTC/Attention hamda tranformer arxitekturasi haqida qisqacha ma’lumot berilgan. Nutq signalini matnga o‘tkazishda so‘nggi yillarda keng foydalanilayotgan neyron tarmog‘i arxitekturalari modellari va transformer arxitekturasiga asoslangan neyron tarmog‘i modeli yaratilgan o‘zbek tili nutq korpusi asosida o‘qitildi va olingan natijalar qiyosiy tahlil qilingan.*

Kalit so‘zlar: transformer arxitekturasi, mashinaviy o‘qitish, neyron tarmog‘i, yashirin holat ifodasi, maqsad vektori, rekurrent neyron tarmoq.

Аннотация. *В этой статье представлены проблемы, возникающие при машинном обучении распознавания речи, и методы глубокого обучения, используемые для их решения. Кроме того, представлен обзор перехода к архитектуре кодер-декодер, основанной на механизме внимания. Также представлен краткий обзор архитектур CTC/Attention и трансформер. При преобразовании речи в текст на основе речевого корпуса узбекского языка были обучены широко используемые в последние годы модели нейросетевой архитектуры и нейросетевая модель, основанная на архитектуре трансформер, и проведено сравнение полученных результатов.*

Ключевые слова: Архитектура трансформер, машинное обучение, нейронная сет, выражение скрытого состояния, вектор сели, рекуррентная нейронная сет.

Abstract. *This article presents the challenges faced in machine learning speech recognition and the deep learning methods used to address them. It also provides an overview of the transition to an attention-based encoder-decoder architecture. A brief overview of the CTC/Attention and Transformer architectures is also provided. Using a speech-to-text corpus of the Uzbek language, we trained widely used neural network architecture models and a neural network model based on the Transformer architecture, and compared the results..*

Keywords: Transformer architecture, machine learning, neural network, hidden state expression, goal vector, recurrent neural network.

Kirish. Mashinaviy o‘qitish/chuqur o‘qitish bilan hal etiladigan masalalarning katta qismida kirish ma’lumotlariga mos bo‘lgan oldindan berilgan chiqish natijalari to‘plami mavjud bo‘ladi. Ushbu kirishlarning barchasini chiqishlarga mos kelishini aks ettiradigan ma’lum bir $f()$ funksiyasi mavjud deb olinadi. Nutq tanib olishda belgilar vektorlari nutq signalining qisqa vaqt spektrlari asosida, ya’ni har 20-25 ms yoki shunga yaqin vaqt oralig‘i uchun belgilar vektorlari shakllantiriladi.

Nutqni tanib olishdagi asosiy muammolardan biri bu bitta tovush (fonema) bilan boshqasi o‘rtasida chegarani oldindan ma’lum emasligidir. Agar nutq signalini qayta ishlash dasturidan foydalanib ~200-300 msgacha qisqartish orqali eshitilsa, uni tushunib bo‘lmaydi.

O‘qitish bazasi nutq signali va unga mos so‘zlar ketma-ketligidan iborat bo‘ladi. Bunda so‘z (yoki fonema)ning tugashi va boshlanishi haqida ma’lumotlar saqlanmaydi.

Ayrim xatolik funksiyalari (kross-entropiya, o‘rtacha kvadratik xatolik) neyron tarmog‘ining kirish va chiqish qismlarining birga-bir mosligini talab qiladi. Biroq, ba’zi masalalarda bunday moslik mavjud bo‘lmaydi. Masalan, nutqni tanib olish, mashinaviy tarjima, qo‘lyozma matnlarni tanib olish kabilar. Bunday hollarda o‘zaro moslikni o‘rnatish CTC, diqqat mexanizmi, seq2seq modellari orqali amalga oshiriladi.

Asosiy qism. To‘liq neyron tarmoqlarga asoslangan ko‘plab tizimlar kodlovchi tarmoq moduliga ega bo‘ladi. Agar ikki yo‘nalishli LSTM kodlovchisi olinsa, u holda oldingi yashirin holat belgilar vektorlari ketma-ketligini olinadi va ma’lum bir kodlovchi funksiyasidan foydalangan holda ularni vektorlarning yashirin (kodlangan) ketma-ketligiga o‘zgartiriladi.

Har bir kodlangan h_t tavsif o‘zida kiruvchi ketma-ketlikdagi t-kirishga fokuslanib umumiy ketma-ketlik to‘g‘risidagi ma’lumotlarni saqlaydi.

$$h_t = Encoder\left(x_t, h_{t-1}\right) \qquad (1)$$

(1) ifoda t vaqt momentidagi yashirin holat ifodasi bo‘lib, Encoder()-bu yashirin tavsifni yangilashda kodlashni amalga oshiruvchi ma’lum bir funksiya.

Odatda bu turdagi kodlovchilar bir nechta qatlamli bo‘lishi mumkin, ya’ni har bir keyingi qatlam oldingi qatlamlarning chiqishini o‘zgartiradigan chuqur BLSTM kodlovchiga ega bo‘ladi. Nutqni tanib olishda ikki yo‘nalishli tarmoqdan foydalanish maqsadga muvofiq hisoblanadi, chunki so‘zni qanday talaffuz qilinishi oldin keluvchi va keyingi fonemalarga ham bog‘liqdir. Ko‘plab undosh tovushlar o‘zidan keyin unli yoki undosh tovush kelishiga bog‘liq holda turlicha talaffuz qilinadi. Masalan, “talaba” va “pasport” so‘zlaridagi “t” harfi. Bu kabi hollar uchun ikki yo‘nalishli LSTM kodlovchisidan foydalanish nutq signalini yaxshiroq modellashtirish imkonini beradi.

LSTM kodlovchi ham xuddi boshqa chuqur neyron tarmoqlari kabi kirishda ma’lum bir x vektorni qabul qilib chiqishda y vektorni beradi biroq, bu vektor to‘liq kirish qiymatiga bog‘liq bo‘lmaydi va u ma’lum darajada oldingi (ikki yo‘nalishli bo‘lsa keyingi) ketma-ketlik qiymatlariga ham bog‘liq bo‘ladi.

Connectionist Temporal Classificationda tekislash mumkin bo‘lgan barcha vaqt-simvol moslashtirishlar integratsiyasi orqali amalga oshiriladi.

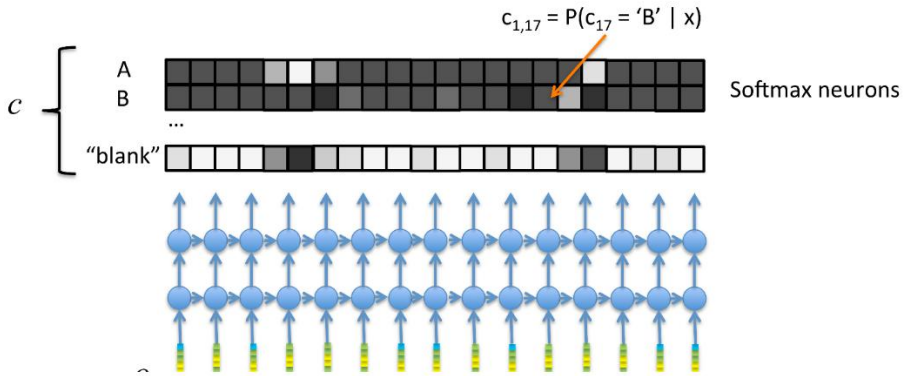
$$L_{ctc}\left(X, W\right) = \sum_{C:k(C)=W} p\left(C \mid X\right)$$

$$L_{ctc}\left(X, W\right) = \sum_{C:k(C)=W} \prod_{t=1}^{T} p\left(c_t \mid X\right)$$

Masalan, "salom" so‘zi olinganda, modelning chiqishida 6 ta maqsad vektori hosil bo‘ladi. Ushbu vektorlarni beshta harf bilan "salom" ko‘rinishida ifodalash masalasi qo‘yiladi. Dastlab CTC xaritalash uchun qancha usul borligini hisoblab chiqadi. Ushbu algoritmda mumkin bo‘lgan $k\left(C\right) = W$: saalom, saloom, _salom, sa_lom, salom_ kabi bir nechta variantlar to‘plamidan eng katta ehtimollikka egasini aniqlashga imkon beradi. Bu yerda "_" – bo‘sh joy belgisi.

LSTM kodlovchi maqsad vektorining har bir o‘lchovi joriy qadamida paydo bo‘lish ehtimoliga mos keladi. 1- rasmda 17-qadamda "B" harfi paydo bo‘lish ehtimolini ko‘rish mumkin.

CTC ehtimolini hisoblash har bir mumkin bo‘lgan simvollar ketma-ketligini sanash va ularni qo‘shish orqali amalga oshiriladi. Agar "Salom" so‘zi o‘qitilsa, unga mos keladigan simvollar ketma-ketligining alohida ehtimolliklari hisoblanadi va ularning yig‘indisi olinadi. Bunda ushbu parametr qiymatini maksimallashtirish talab etiladi.



1-rasm. Kodlovchi chiqishi

Simvollarning to‘g‘ri ketma-ketligi ehtimolligini maksimallashtirish uchun tarmoq parametrlari quyidagi formula orqali yangilanadi.

$$\theta^* = \arg\max_{\theta} \sum_i \log P\left(y^{*(i)} \mid x^i\right) = \arg\max_{\theta} \sum_i \log \sum_{c:\beta(c)=y^*(i)} P\left(c \mid x^i\right)$$

Encoder-Decoder tuzilmasida Encoder butun kirish ketma-ketligini o‘zgarmas o‘lchamli h_t vektorda jamlashga harakat qiladi. Kodlovchi sifatida o‘zi har bir kirish belgilar vektori x_t ni qabul qiluvchi va ichki holatini h_t ichida shu vaqtgacha ketma-ketlikni ko‘rsatish (yig‘ish) uchun xizmat qiluvchi rekurrent neyron tarmoq (RNN / LSTM / BLSTM / GRU) dan foydalaniladi.

Bashoratlash uchun har bir qadamda h_t ni olish mumkin, lekin T vaqtdagi ketma-ketlikning oxirigacha kutish va chiqish ketma-ketligini generatsiya qilishni boshlash uchun h_T tavsifni olish maqsadga muvofiq hisoblanadi. Buning sababi, so‘z/harf/fonemaning chegaralari aniq emas va Encoder kirish ketma-ketligini h_T ichida to‘liq umumlashtirishi yaxshiroq natija beradi.

Kirish sifatida <**sos**> - ketma-ketlikning boshlanishi tokeni dekodlovchiga beriladi va chiqish simvollarini generatsiyasi boshlanadi. dekodlovchi - bu natijani taxmin qilish uchun har safar ichki holatini o‘zgartiradigan yana bir rekurrent neyron tarmoq (ikki yo‘nalishli emas).

Har bir vaqt qadamida, joriy chiqishni taxmin qilish uchun oldingi vaqt qadamidagi chiqishdan foydalaniladi, ya’ni

$$s_i = Decoder\left(s_{i-1}, y_{i-1}\right). \qquad (2)$$

(2) formula $i-$ simvolni bashorat qilishdagi dekodlovchining yashirin holatini ifodalaydi. Bu yerda $Decoder(\ )$ o‘zining yashirin ichki holatini yangilash uchun LSTM dekodlovchidan foydalanuvchi ma’lum bir funksiya hisoblanadi.

Kodlovchi/dekodlovchi tarmog‘ining ishlashiga doir misol 2- rasmda keltirilgan.

Dekodlovchi <eos> - ketma-ketlikning oxiri tokenini generatsiya qilganda chiquvchi simvollar ketma-ketligini generatsiya qilish to‘xtatiladi.

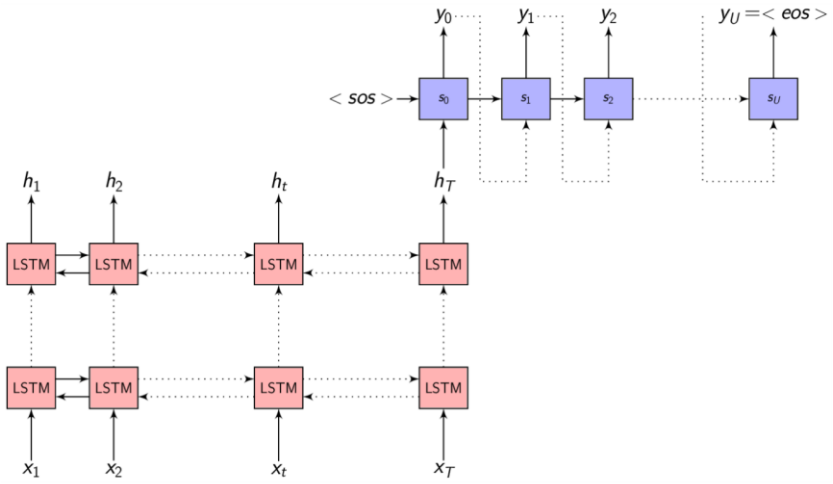
Dekodlovchining $s(i-1)$ yashirin holatini (oldingi chiqish vaqtidagi) va $y(i-1)$ chiqish simvolini (oldingi chiqish simvolini) hisobga olgan holda, joriy vaqt bosqichida chiqish simvolini quyidagicha bashorat qilish mumkin:

$$p\left(y_i \mid \{y_1, y_2, ..., y_{i-1}\}\right) = g\left(y_{i-1}, s_i\right), \quad (3)$$

bu yerda $g(\ )$ - dekodlovchi funksiyasi.

To‘liq chiqish ketma-ketligi y ning ehtimolligi quyidagi formula orqali hisoblanadi:

$$p(y) = \prod_{i=1}^{U} p\left(y_i \mid \{y_1, y_2, ..., y_{i-1}\}, s_i\right) \quad (4)$$



2-rasm. Kodlovchi-dekodlovchi

Kodlovchi-dekodlovchi bilan bog‘liq quyidagi muammolar mavjud:

- neyron tarmog‘i belgilar vektorlari kirish ketma-ketligining barcha muhim axborotlarini o‘zgarmas o‘lchamli vektorga siqish imkoniyatiga ega bo‘lishi kerak;
- agar ketma-ketlik uzun bo‘lsa, ayniqsa testlashda kirish ketma-ketligi o‘qitish ketma-ketligidan ancha uzun bo‘lsa, u holda kodlovchi/dekodlovchi tarmog‘ining ishlash samaradorligi yomonlashadi;
- kodlovchining belgilar vektorlarining butun ketma-ketligini o‘zgarmas o‘lchamli vektorga yig‘ishi vektorning o‘lchamiga bog‘liq (jumla qancha uzun bo‘lsa, vektor ham shuncha uzun bo‘lishi) maqsadga muvofiq, lekin buni ushbu tarmoqda amalga oshirib bo‘lmaydi, chunki ketma-ketlikning uzunligi sezilarli darajada farq qilishi mumkin.

Yuqorida sanab o‘tilgan muammolarni hal qilishda tavsiya etiladigan usullaridan biri bu diqqat (Attention) mexanizmidan foydalanish hisoblanadi. Diqqat mexanizmi kodlovchisi bu - dekodlovchi freymvorkining kengaytmasi hisoblanadi.

Model chiqish simvolini generatsiyalashi zarur bo‘lganda, u (dasturiy jihatdan) eng muhim ma’lumot to‘plangan belgilar vektorlari kirish ketma-ketligidagi pozitsiyalar to‘plamini izlaydi va to‘plamni aniq tanlashi muhim hisoblanadi. Kodlovchi-dekodlovchi freymvorkidan asosiy farqi butun kirish ketma-ketligini o‘zgarmas o‘lchamli vektorga yig‘ish shart emasligi hisoblanadi. Bunda h_T ning yashirin holatini taqdim etish o‘rniga, dekodlovchilar tarmog‘iga chiqishni generatsiyalash uchun ma’lum bir kontekstga eng mos keladigan h ning qism to‘plami tanlanadi. Kontekst vektori deb ataluvchi C_i vektorni hosil qilish uchun ushbu relevant h_T chiziqli ravishda o‘zgartiriladi.

$$C_i = q\left(\{h_1, h_2, ..., h_T\}, \alpha_i\right) \quad (5)$$

Diqqat mexanizmi qaysidir ma’noda joriy kontekstga eng mos keladigan kirish belgilarining qism to‘plamini hisobga oladigan modeldagi mexanizm hisoblanadi. Chuqur o‘qitish usullarining barchasida xatoliklarni teskari tarqatish algoritmi orqali o‘qitish uchun funksiyalar differensiallanuvchi bo‘lishini kerak. Ushbu differensiallanuvchi qism to‘plamga Diqqat mexanizmini qo‘llash uchun barcha kirish belgilar vektorlariga har xil vaznlar bilan diqqat qaratiladi.

Diqqat mexanizmi kodlovchi-dekodlovchidan asosiy farqi quyidagilardan iborat.

- yuqorida muhokama qilgan kodlovchi-dekodlovchi tarmog‘ida Dekodlovchining yashirin holati quyidagicha hisoblanadi:

$$s_i = f\left(s_{i-1}, y_{i-1}\right).$$

- Diqqat mexanizmi kengaytmasida Dekodlovchining yashirin holatini hisoblashda kontekst vektori inobatga olinadi:

$$s_i = f\left(s_{i-1}, y_{i-1}, C_i\right).$$

Kontekst vektori o‘zida faqatgina eng relevant kirish belgilar vektorini aks ettiradi. Ushbu relevantlikni baholash uchun α o‘zgaruvchini ko‘rib chiqish zarur. Bunda α_i kontekst vektor C_i tarkibidagi h_i ko‘rinishida kodlangan yashirin holatning vazni bo‘lib, i - vaqt momentidagi chiqishni bashoratlash uchun xizmat qiladi. α ni inobatga olgan holda, kontekst vektor C_i ni quyidagicha hisoblash mumkin:

$$C_i = \sum_{j=1}^{T} \alpha_{i,j} \cdot h_j$$

$$\sum_{j=1}^{T} \alpha_{i,j} = 1$$

- $\alpha_{i,j}$ ni hisoblash uchun $i-$ simvolni bashoratlashda j- izoh vektorining muhimlik darajasi bo‘lgan $e_{i,j}$ ni aniqlash kerak. Har bir izoh h_j ning $\alpha_{i,j}$ vazni quyidagicha hisoblanadi.

$$\alpha_{i,j} = Soft\max\left(e_{i,j}\right) = \frac{e^{e_{i,j}}}{\sum_{k=1}^{T} e^{e_{i,k}}}$$

$$\sum_{j=1}^{T} e_{i,j} \neq 1$$

bu yerda $e_{i,j} = AttentionFunction\left(s_{i-1}, h_j\right)$ - bu dekodlovchining $s\left(i-1\right)$ yashirin holati bilan har bir h_j izohning muhimligini hisoblaydigan moslik funksiyasi.

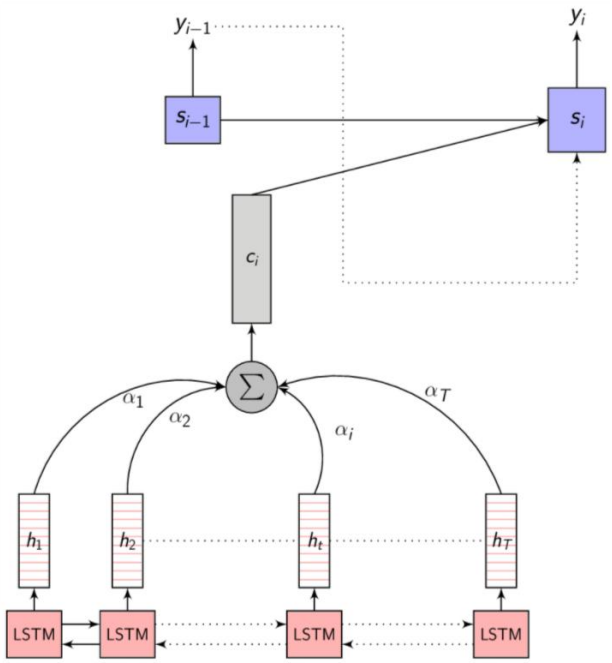
Modelni o‘qitishdan oldin, ya’ni 0-epoxada diqqat mexanizmi vaznlari qiymatlari tasodifiy bo‘ladi va shuning uchun C_i kontekst vektori ahamiyatsiz kirish belgilar vektorlaridagi keraksiz xalaqitni o‘z ichiga oladi. Bu esa model samaradorligini pasaytiradi. Yaxshi diqqat modeli yaxshi kontekst vektorini taqdim etadi bu esa model samaradorligi oshishiga olib keladi.

Abstrakt shaklda buni quyidagicha ko‘rsatish mumkin. Belgilar vektorlarining har bir ketma-ketligi ma’lum vazn bilan aralashtiriladi va keyin qaror qabul qilish uchun dekodlovchi tarmog‘iga uzatiladi. Belgilar va ularga biriktiraladigan vaznlar AttentionFunction tomonidan hal qilinadi.

Model qanday o‘qitilishini chuqurroq anglash uchun diqqat mexanizmi vaznlari vaqt o‘tishi bilan (epoxalarda) qanday o‘zgarishini ko‘rish mumkin. Ayrim adabiyotlarda ko‘plab diqqat mexanizmiga asoslangan modellari bayon etilgan. ESPnet kabi vositalar to‘plamida tajribiy tadqiqotlarda oson foydalanish mumkin bo‘lgan 10 dan ortiq diqqat mexanizmlari mavjud.

Diqqat mexanizmiga asoslangan nutqni avtomatik tanib olish tizimlari moslashuvchan tekislash xususiyati tufayli o‘chirish va kiritish xatolariga moyil bo‘ladi. Bunga keyingi simvolni bashorat qilish uchun kodlovchi holatlari ketma-ketligining ixtiyoriy qismini hisobga olish sabab bo‘ladi. Diqqat dekodlovchi tarmog‘i tomonidan generatsiya qilinganligi sababli, u barcha kodlovchi kadrlariga ahamiyat bermagan bo‘lsa ham, ketma-ketlikning tugashini oldinroq taxmin qilishi mumkin. Bu jumlani juda qisqa bo‘lishiga olib keladi. Boshqa tomondan, u avvalgi qismlar bilan bir xil qismlarga e’tibor berib, yuqori ehtimollik bilan keyingi simvolni taxmin qilishi mumkin. Bunday holda, gipoteza juda uzun bo‘ladi va bir xil simvollar ketma-ketligini takrorlashi mumkin. Bu va boshqa ko‘plab sabablarga ko‘ra, keltirilgan ikki arxitekturaning afzalliklarini o‘qitish va dekodlashda foydalanadigan

hamkorlikdagi gibrid CTC/Attention arxitekturasi taklif qilingan. O‘qitish davomida ishonchlilikni oshirish va tezkor yaqinlashishga erishish uchun ko‘p mezonli o‘qitishdan foydalaniladi. Dekodlash paytida notekis taqsimlanishni kamaytirish uchun diqqat mexanizmi asosidagi va CTC-baholarni birlashtirib bir martalik beam search algoritmida qo‘shma dekodlash amalga oshiriladi.



3-rasm. Diqqat mexanizmining abstrakt tasvirlanishi

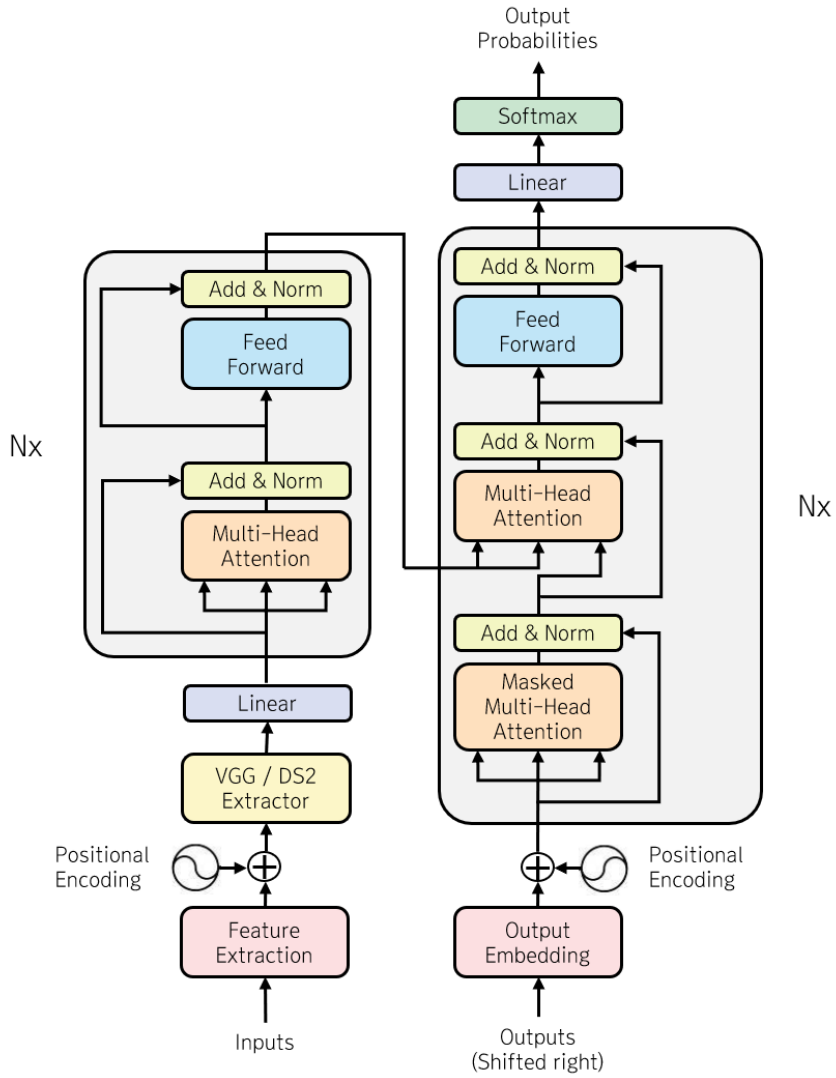
Speech-Transformer — ASR uchun recurrence (RNN) dan voz kechib, to‘liq attention mexanizmlariga tayanuvchi seq2seq model. U dastlab Dong va hammualliflar tomonidan ICASSP 2018 ishida “no-recurrence” ASR modeli sifatida taqdim etilgan.

Ushbu modelning asosiy g‘oyasi nutq signali (odatda log-Mel filterbank kabi belgilar) uzun ketma-ketlik bo‘lgani uchun, model vaqt bo‘ylab global kontekstni self-attention orqali ko‘radi, encoder–decoder arxitekturasi orqali “akustika → matn” xaritalashni end-to-end shaklda o‘qitiladi hamda o‘qitish GPUda samarali parallel amalga oshiriladi.

Transformer arxitekturasi aslida “Attention Is All You Need” ishida ta’riflangan encoder–decoder tamoyiliga tayanadi. Speech-Transformer ham siz keltirgan matndagi kabi encoder va decoder bloklaridan tuziladi, ammo RNN o‘rniga Transformer qatlamlari ishlatiladi.

Encoder kirish nutq belgilari ketma-ketligini T uzunlikda qabul qiladi va uni yuqori darajadagi kontekst vektorga aylantiradi. Har bir encoder qatlami odatda Multi-Head Self-Attention, Position-wise Feed-Forward Network, qatlam bo‘yicha normallashtirish, kabi qismlardan tashkil topadi. Nutq signalida T uzunlik juda katta bo‘lishi mumkin, shuning uchun Speech-Transformerda vaqt–chastota strukturasini yaxshiroq inobatga olish uchun 2D attention g‘oyalari ham muhokama qilingan.

Decoder matn birliklarini avto-regressiv tarzda hosil qiladi. Masked self-attention bilan ketma-ketlik tarixini, oldingi tokenlarni ko‘radi hamda kelajak tokenlarni “ko‘rmaslik” uchun maska ishlatiladi. Keyin encoder–decoder attention orqali encoder chiqishidan (nutq kontekstidan) kerakli joylarni tanlab oladi. So‘ng feed-forward va softmax orqali $p(y_u \mid y_{<u}, x)$ ni beradi. Bu “maska” g‘oyasi Transformer decoderning standard komponenti bo‘lib, seq2seq generatsiyada sababiylikni saqlaydi.



4-rasm. Speech-Transformer arxitekturasi

Taklif etilgan neyron tarmoq modelini o‘qitish uchun “Ma’lumotlarga ishlov berish tizimlari” laboratoriyasida o‘zbek tilidagi nutq korpusi yaratildi. Undagi nutqlar davomiyligi 364 soatdan iborat audiokitoblardan va 72 soatli aforizm hamda maqollar audioyozuvidan iborat. Bazani yaratishda 168 ta suxandonlar qatnashgan bo‘lib, neyron tarmoq modelini o‘qitishda dastlab qisqa davomiylikka ega (3-10 sekund) audio fayllar ketma-ketligidan foydalanilgan. Model yaqinlashishi ortgandan so‘ng 6-9 daqiqagacha davomiylikdagi audio fayllar ketma-ketligi o‘qitish uchun berildi. Audio fayllar 16 kHz diskretlash chastotasiga ega bo‘lib, wav, mp3 formatlarida saqlangan. Har bir audio fayl uchun mos bo‘lgan matn fayl mavjud. Nutq bazasining umumiy hajmi 46 Gb ni tashkil etadi. Neyron tarmog‘i modelini o‘qitishda nutq bazasining 90% qismidan foydalanilgan va qolgan 10% qismi o‘qitilgan modelni testlash jarayonida qo‘llanilgan.

Tajribaviy tadqiqotlar. Neyron tarmoqlari modellarini o‘qitishda va tajribaviy tadqiqotlarni o‘tkazishda i9 protsessor, 32 GB operativ xotira, RTX 3090 Ti videokartali kompyuterdan foydalanildi. Neyron tarmoqlari modellarini qurish va amalga oshirishda Python dasturlash tili hamda Pytorch chuqur o‘qitish freymvorkidan foydalanilgan. Birinchi model 2 ta CNN qatlam, 3 ta LSTM qatlam va o‘qitishda CTC loss funksiyasidan foydalanildi. Ikkinchi model 2 ta CNN qatlam, 3 ta BLSTM qatlam va o‘qitishda CTC loss funksiyasidan foydalanildi. Taklif etilayotgan modelda 2 ta BLSTM qatlam audioni kodlovchi qism sifatida foydalanildi. Kodlovchi chiqishida diqqat mexanizmi– Attention va o‘qitishda CTC loss funksiyasidan foydalanildi.

1-jadval. Tajribaviy tadqiqot natijalari

Neyron tarmoq arxitekturasi	O‘qitish vaqti (soat)	O‘qitish aniqligi (%)	Testlash aniqligi (%)	Testlash xatoligi (WER) (%)
CNN+LSTM+CTC loss	12	89	74	26
CNN+BLSTM+CTC loss	15	90	76	24
Transformer based	14	92	79	21

Xulosa. Nutqni mashinaviy o‘qitish orqali tanib olishda paydo bo‘ladigan muammolar va ularni hal qilishda foydalaniladigan chuqur o‘qitishga asoslangan usullar o‘rganildi. Bundan tashqari, diqqat mexanizmiga asoslangan kodlovchi-dekodlovchi arxitektura tizimiga o‘tish haqida umumiy ma’lumot beriladi va an’anaviy arxitekturalarning ba’zi kamchiliklari va ularni Encoder-Attention-Decoder arxitekturasi bilan qanday bartaraf etish usullari taklif etildi. Shuningdek, gibrid CTC/Attention arxitekturasi haqida qisqacha ma’lumot berilib, nutqni matnga o‘tkazishda so‘nggi yillarda keng foydalanilayotgan neyron tarmoqli modellari va diqqat mexanizmiga asoslangan neyron tarmog‘i modeli yaratilgan o‘zbek tili nutq korpusi asosida o‘qitildi va olingan natijalar qiyosiy tahlil qilindi.

Foydalanilgan adabiyotlar

1. Bahl, L. R., Jelinek, F. and Mercer, R. L. A Maximum Likelihood Approach to Continuous Speech Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 1983, vol. PAMI-5, no. 2, p. 179–190.

2. Noisy channel model. [online], 2020, page was last edited on 12 April 2020, at 09:02 (UTC). https://en.wikipedia.org/wiki/Noisy_channel_model

3. Watanabe, S., *et al.* Hybrid CTC. *Attention Architecture for End-to-End*. 2017, vol. 11, no. 8, p. 1240–1253.

4. Hannun “Sequence Modeling with CTC”. [online], 2017, Distill. https://distill.pub/2017/ctc/

5. Yuanyuan Zhao, Jie Li, Xiaorui Wang, and Yan Li. "The SpeechTransformer for Large-scale Mandarin Chinese Speech Recognition." ICASSP 2019.

5. Loubser A, Villiers P, Freitas A “End-to-end automated speech recognition using a character based small scale transformer architecture” // Expert Systems with Applications Volume 252, 15 October 2024