

**ԽՈՐԸ ՈՒՍՈՒՑՄԱՆ ԿԻՐԱՌՈՒԹՅՈՒՆԸ
ԲՈՒՍԱԲՈՒԾՈՒԹՅԱՆ ՃՅՈՒՂՈՒՄ**

ՀՏԴ 004.8

DOI: 10.56246/18294480-2022.13-09

ԱԲԳԱՐՅԱՆ ՀԵՆՐԻԿ

ՀԱԱՀ ագրոբիզնեսի և տնտեսագիտության ֆակուլտետի ուսանող
էլիոստի՝ henri.abgaryan@gmail.com

ՄԱՆՈՒԶԱՐՅԱՆ ԴՈՆԱՐԱ

ՀԱԱՀ տվյալագիտության ամբիոնի դասախոս
էլիոստի՝ mdon2008@yandex.ru

Խորը ուսուցումը (խՈԻ) արհեստական բանականության կիրառություն է, որը համակարգիչներին թույլ է տալիս սովորել և կատարելագործվել: Հսկայական քանակությամբ տվյալներ հաշվարկելու և վերլուծելու իր անհավասարի ներուժով խորը ուսուցումը առաջադեմ գործիք է բարդ առաջադրանքներն ավելի արագ և արդյունավետ կատարելու համար: Խորը ուսուցման հիմնական կենտրոնացումը համակարգչային ծրագրերն են, որոնք կարող են հեշտությամբ ստանալ անհրաժեշտ տվյալները և օգտագործել դրանք սովորելու նպատակով: Տվյալները կարող են լինել վիճակագրական վերլուծություններ, անձնական տվյալներ և այլ նմանօրինակ տեղեկություններ ցանկացած ոլորտից և ցանկացած քանակի: Խորը ուսուցումն ունի բազմաթիվ կիրառություններ:

Սույն հոդվածը ծանոթացնում է խորը ուսուցման և դրա կարևորագույն բաղադրամասերից մեկին՝ պատկերների դասակարգմանը, որը մեքենայական ուսուցման ենթաճյուղերից է: Հոդվածում ներկայացված են խորը ուսուցման համառոտ պատմությունը, պատկերների դասակարգման տարբեր մոդելները, մասնավորապես նեյրոնային ցանցերը, գործնական կիրառությունը ագրարային ոլորտում՝ մանրամասն նկարագրություններով, գրաֆիկներով, ծրագրային կոդ Python լեզվով:

Բանալի բառեր՝ արհեստական բանականություն, խորը ուսուցում, նեյրոնային ցանցեր, տվյալների բազա, Kaggle, Python, Google Colabrotory, փաթեթային ցանց (Convolutional Neural Network), TensorFlow:

Խորը ուսուցումը արհեստական բանականության կիրառումն է: Այն հնարավորություն է տալիս համակարգիչներին առանց նախօրոք ծրագրավորված լինելու ավտոմատացված կերպով սովորելու և բարելավելու արդյունքները: Խորը ուսուցման հիմնական կենտրոնացումը համակարգչային ծրագրերն են, որոնք կարող են հեշտությամբ ստանալ անհրաժեշտ տվյալներ և դրանք օգտագործել սովորելու նպատակով: Իսկ ո՞րն է սովորելու ընթացքը: Ծրագիրը սովորելու կամ «մարզման» գործընթացն սկսվում է տվյալների վերլուծությամբ (Expert.ai Team, 2020): Տվյալները կարող են լինել վիճակագրական վերլուծություններ, անձնական տվյալներ և այլ նմանօրինակ տեղեկություններ ցանկացած ոլորտից և ցանկացած քանակի: Այս տվյալները վերլուծելու նպատակն է գտնել հստակ օրինաչափություններ և նմանություններ, որոնց կիրառությամբ էլ հետագայում համակարգիչը կկարողանա որոշումներ կայացնել: Խորը ուսուցման առաջնային նպատակն է թույլ տալ համակարգիչներին ինքնուրույն սովորել հետազոտությունների օգնությամբ և ըստ այդմ որոշել հետագա գործողությունների հաջորդականությունը առանց մարդկային միջամտության կամ օգնության: Սակայն կարևոր է հասկանալ նաև, որ դասական խորը ուսուցման ալգորիթմները, օրինակ՝ տեքստը, համարում են որոշակի տառերի հաջորդականություն: Որպեսզի հնարավոր լինի հասկանալ բառերի իմաստն ու կիրառության մանրամասնությունները, հարկավոր է ուսումնասիրել, օրինակ, խոսողի դեմքի արտահայտությունը: Այդ իսկ պատճառով տարիների ընթացքում խորն ուսուցման ալգորիթմներն անցել են բազմաթիվ փորձարկումներ և փոփոխություններ, որոնք արդեն իսկ հնարավորություն են ընձեռում համակարգիչներին կատարել այնպիսի հետազոտություններ և պլանավորել այնպիսի քայլերի հաջորդականություն, որը մոտ է մարդկայինին: Խորը ուսուցման կիրառության ոլորտներից են որոնողական համակարգերի, պատկերների ճանաչման, կենսահնֆորմատիկայի, ձեռագրերի ճանաչման և այլ բնագավառներ:

Պատկերների ճանաչումը պատկերները ճանաչելու և դրանք ճիշտ դասակարգելու խնդիր է: Վերջինիս կարևորությունն այն է, որ վարժեցված համակարգչային ծրագիրն ի զորու է վերլուծելու միլիոնավոր պատկերներ հարաբերականորեն կարճ ժամանակահատվածում, որը մարդու համար կարող է անհրաժեշտ լինել: Խորը ուսուցման մեջ պատկերների ճանաչումը լայն տարածում ունի, և դրա լուծումները ներառում են խորը ուսուցման գրեթե բոլոր ալգորիթմները (altexsoft 2019): Սակայն մինչ այժմ կատարված փորձարկումներն ու արդյունքների համեմատությունները ցույց են տալիս,

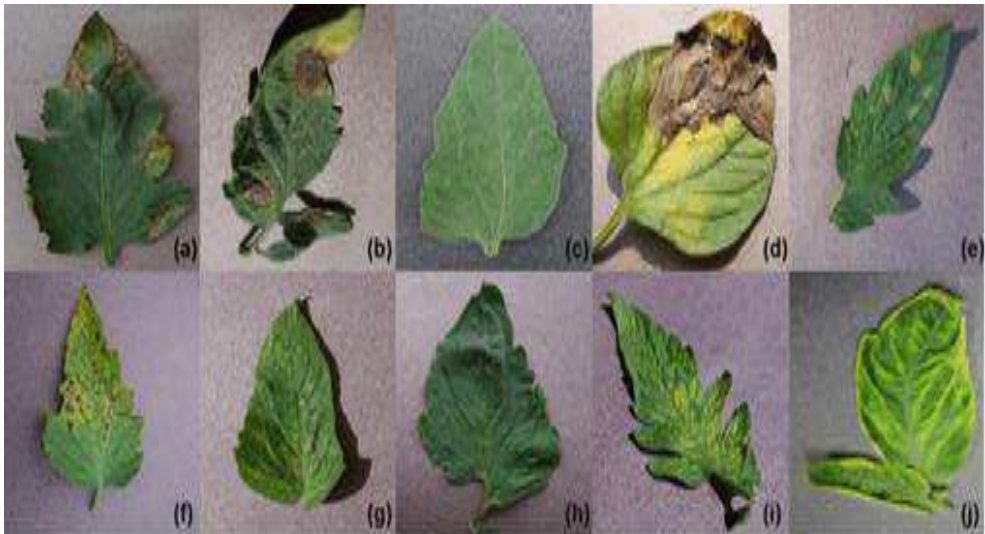
որ ամենաարդյունավետ ալգորիթմները ներդրումնային ցանցերով կառուցվածներն են: Ներդրումնային ցանցերը որոշակի քայլերի (շերտեր) հաջորդականություն են, որոնցից յուրաքանչյուրի ժամանակ տվյալները ենթարկվում են նախօրոք մշակված ֆունկցիայի ազդեցությանը: Պատկերների ճանաչման կամ դասակարգման խնդրում այս շերտերից ամեն մեկը պատասխանատու է նախորդ շերտերում կատարված արդյունքի վրա նոր փոփոխություն կատարելու համար: Ինչպես, օրինակ, առաջին շերտը կարող է պատասխանատու լինել գունավոր նկարը սև ու սպիտակի փոխակերպելու համար, երկրորդ շերտն այդ արդեն իսկ սև ու սպիտակ արդյունքի վրա կարող է ավելացնել դեղին կետեր, և այդպես շարունակ: Այս գործընթացը շարունակվում է այնքան ժամանակ, մինչև տվյալների բազայում առկա բոլոր պատկերներն անցնում են ներդրումնային ցանցով, և բոլորի համար առկա են լինում հստակ պատասխաններ: Ներդրումնային ցանցերի կիրառությունը պատկերների ճանաչման խնդրում արդյունավետ է, քանի որ կարիք չկա նախօրոք վերամշակելու և փոփոխելու պատկերները:

Մեր կողմից ներկայացվող արհեստական բանականության, պատկերների դասակարգման վրա հիմնված ծրագիրը հնարավորություն է տալիս դասակարգել լուրիկի 9 ամենահաճախ հանդիպող հիվանդությունները: Առաջարկվող ծրագիրը հիմնված է Նավին Մոնդադեյի կողմից ստեղծված «Identifying-Nine-Tomato-Disease-With-Deep-Learning» ծրագրի վրա: Չնայած նրան, որ մեր ծրագրի հիմքում վերոնշյալ ալգորիթմն է, մենք այն միացրել ենք Telegram հավելվածին՝ բոլորին հասանելի դարձնելու համար: Ներդրումնային ցանցը մարզելու ամենավերջին փուլում, երբ արդեն ներդրումնային ցանցի մոդելը պատրաստ է, մենք ստեղծում ենք Telegram Bot և մոդելը միացնում ենք Bot-ին: Ծրագրում օգտագործվում է Google-ի կողմից ստեղծված Kaggle տվյալների շտեմարանից «PlantVillage» տվյալների բազան, որն իր մեջ ներառում է շուրջ 20 հազար նկար՝ ներդրումնային ցանցը «մարզելու» համար: Ծրագրում օգտագործում ենք Google-ի կողմից ստեղծված Kaggle համակարգը, որն իր մեջ ներառում է անհրաժեշտ տվյալների բազան ներդրումնային ցանցերը «մարզելու» համար: Kaggle-ը տվյալագետների և մեքենայական ուսուցման պրակտիկայով զբաղվողների առցանց համայնք է:

Ինչպես նշվեց, պատկերների ճանաչումը կատարվում է ներմուծված պատկերը ներդրումների շերտերի միջով անցկացնելիս. այսպես, առաջին շերտում կատարվում է պատկերի ֆոնային փոփոխություն, միջին շերտերում կատարվում է օբյեկտի ձևի ստուգում, իսկ արդեն վերջին շերտերում կա-

տարվում է վերջնական օբյեկտի ճանաչում: Ծրագրի աշխատանքի գործընթացը կներկայացնենք 5 փուլերի օգնությամբ: Որպես ծրագրի գործարկման միջավայր (runtime environment)՝ օգտագործելու ենք Google Colabrotory-ին, որը հնարավորություն է տալիս աշխատեցնել Python ծրագրավորման լեզուն առցանց տարբերակով հենց Google-ի սերվերների վրա:

Փուլ 1. Մինչև ծրագրի աշխատեցումը անհրաժեշտ է հավաքել համապատասխան տվյալներ, որպեսզի մարզենք նեյրոնային ցանցերը: Kaggle-ում առկա նկարների տվյալների բազայից անհրաժեշտ է բեռնել մեզ անհրաժեշտ տվյալների շտեմարանը: Քանի որ ծրագիրը «գուշակելու» է լուլիկի 9 հիվանդություններից մեկը, մեզ անհրաժեշտ է հավաքել այդ հիվանդություններին համապատասխանող լուլիկի տերևի լուսանկարներ: Դրա համար կստեղծենք նկարների բազա. ինչքան նկարների բազան մեծ լինի, այնքան ավելի լավ, քանի որ մարզումը կկատարվի նկարների ավելի մեծ բազայի վրա, և արդյունքում կունենանք ավելի հստակ պատասխան: Նեյրոնային ցանցերի մարզման այս փուլը հատկապես կարևոր է, քանի որ ճիշտ ընտրված տվյալներն են ծրագրի բարձր արդյունավետության հիմքը: Kaggle-ի տվյալների բազայի օգտագործումը հեշտացնում է առաջին փուլի աշխատանքը: Քանի որ յուրաքանչյուր նկար ունի միևնույն չափսերը՝ 256x256, անհրաժեշտություն չկա նկարները խմբագրելու:



Նկ. 1. Տվյալների բազայից մի քանի օրինակ:

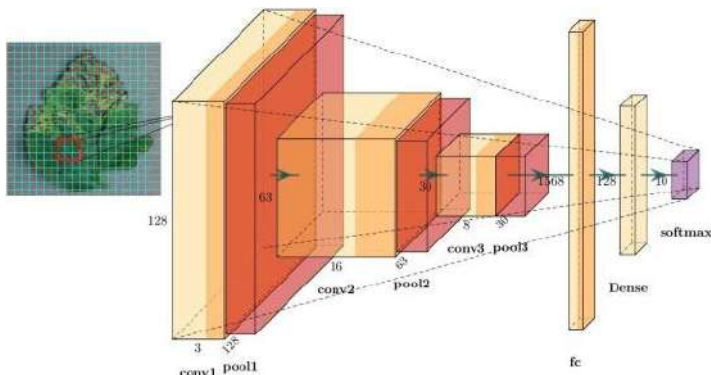
Նկար 1-ում ներկայացված է հավաքագրված նկարների բազայից մի քանի օրինակ: Հարկ է նշել, որ տվյալների բազայում, բացի հիվանդ լուլիկի բույսի տերևների նկարներից, անհրաժեշտ է ունենալ նաև առողջ բույսի տերևների նկարներ: Այսպիսով, արդյունքում ստացվեց դասակարգման 9 տեսակ՝ «բակտերիալ բիծ», «վաղ բորբոս», «առողջ տերև», «ուշ բորբոս», «տերևային սունկ», «սեպտորիա», «իզարիոպսիս», «լուլիկի խճանկարային վիրուս», «լուլիկի դեղին վիրուսային հիվանդություն»:

Տվյալների բազան հավաքելուց հետո անհրաժեշտ է այն բաժանել մարզման (train) և ստուգման (test) համար նախատեսված տվյալների շտեմարանի:

Փուլ 2. Հաջորդ քայլով տվյալները բեռնելուց հետո մենք պետք է յուրաքանչյուր պատկերի փիքսելային արժեքները (0-255) վերափոխենք 0-1 միջակայքի արժեքով, քանի որ նեյրոնային ցանցերը բավականին լավ են աշխատում նորմալացված տվյալների հետ: Փիքսելային արժեքների ամբողջ զանգվածը վերափոխվում է torch tensor-ի, ապա բաժանում 255-ի:

Փուլ 3. Ստեղծում ենք մոդելի կառուցվածքը: Մենք օգտագործելու ենք փաթեթային նեյրոնային ցանց (Convolutional Neural Network), որը ներկայացվել է դեռևս 1990-ական թվականներին: Ավանդական նեյրոնային ցանցերի դեպքում, երբ մուտքային տվյալները նկարներ են, ճիշտ լուծմանը զուգամիտելու համար բավականին երկար ժամանակ է անհրաժեշտ, սակայն փաթեթային նեյրոնային ցանցերի դեպքում այդ ժամանակը մի քանի անգամ պակասում է:

Նկար 2-ում ներկայացված է փաթեթային նեյրոնային ցանցի (Convolutional Neural Network) օրինակ:



Նկ. 2. Փաթեթային ցանցի (Convolutional Neural Network) օրինակ:

Հիմա փորձենք ծրագրավորել նկար 2-ի սխեման համապատասխան ծրագրային իմպլեմենտացիայով (տե՛ս նկ. 3):

```
model = Sequential()

#convolution
model.add(Conv2D(32, (3,3), input_shape = (img_width, img_height, 3), activation='relu', padding='same'))
model.add(MaxPool2D(2,2))

model.add(Conv2D(64, (3,3), activation='relu', padding='same'))
model.add(MaxPool2D(2,2))

model.add(Conv2D(128, (3,3), activation='relu', padding='same'))
model.add(MaxPool2D(2,2))

model.add(Conv2D(192, (3,3), activation='relu', padding='same'))
model.add(MaxPool2D(2,2))

#Dense
model.add(Flatten())

model.add(Dense(128, activation='relu'))
model.add(Dropout(0.4))

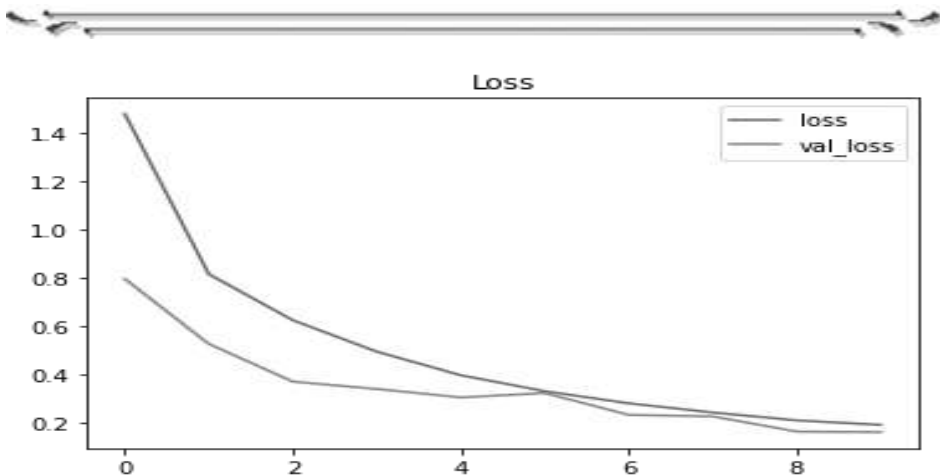
model.add(Dense(256, activation='relu'))
model.add(Dropout(0.3))

model.add(Dense(10, activation='softmax'))
```

Նկ. 3. Փաթույթային ցանցի (Convolutional Neural Network) ծրագրային ներկայացումը TensorFlow-ով:

Փուլ 4. Ստեղծված մոդելի «մարզում»: Մենք կօգտագործենք «մարզման» տեմպի ժամանակացույց (Learning Rate Scheduling) ասվածը, որը յուրաքանչյուր մարզումից հետո կփոխի նեյրոնային ցանցի սովորելու գործակիցը (learning rate): Մարզման ընթացքում սովորելու գործակիցը (learning rate) փոխելու բազմաթիվ ռազմավարություններ կան: Մենք կօգտագործենք հետևյալ ռազմավարությունը՝ սկզբում կընտրենք ցածր սովորելու գործակից (learning rate), իսկ հետո այն խմբաքանակ (batch) առ խմբաքանակ կբարձրացնենք: Մոդելի մարզումը բավականին երկար գործընթաց է, կարող է տևել անգամ մի քանի ժամ:

Փուլ 5. Տեսնենք, թե արդյունքում ինչ ստացվեց: Տվյալների բազայից նախօրոք առանձնացրել էինք նկարներ, որոնք մոդելը դեռ չէր «տեսել», այդ նկարները տեղակայված են վավերացման (val) համար նախատեսված տվյալների շտեմարանում: Մոդելի նպատակն է նախորդ տվյալների հիման վրա «գուշակել» մոտեքագրված նոր տվյալը: Նկար 4-ում կարելի է տեսնել, թե ինչպես է գուշակված տվյալի սխալմունքը (loss) նվազում սովորելու ժամանակին զուգընթաց:



Նկ. 4. «Մարզման» (train) և վավերացման (validation, կրճապ՝ val) փուլերի սխալմունքի գրաֆիկը:

Փուլ 6. Տեսնենք, թե արդյունքում ինչ ստացվեց: Արդեն վերջին փուլում մենք մեր ստացած մոդելը պետք է միացնենք Telegram Bot-ին, դրա համար կրկին կօգտվենք Python ծրագրավորման լեզվից և մի քանի գրադարաններից՝ telegram.ext, io և open-cv: Նախ պետք է ստանալ տեղեկրամյան հատուկ բանալին (token), որը հնարավորություն է տալիս միանալու տեղեկրամի սերվերներին: Այնուհետև մի քանի ֆունկցիաների օգնությամբ (նկար 5) կպատրաստենք հնարավոր ավտոմատ պատասխաններ՝ օգտատերերի հարցերին պատասխանելու համար:

```
def start(update, context):
    update.message.reply_text('Բարի գալուստ 🌟 Այգոլիտ, ձեր գյուղատնտես օգնականը 🤖\n '
                              'Ես դեռևս կարողանում եմ միայն ախտորոշել լոլիկի 🍷 9 տեսակի հիվանդություն 🤒\n'
                              'Որպեսզի իմանաք թե ձեր լոլիկի դաշտը որ հիվանդությունն է վարակել,'
                              'խնդրում եմ ինձ ուղարկել լոլիկի հիվանդ տերևի նկար:')

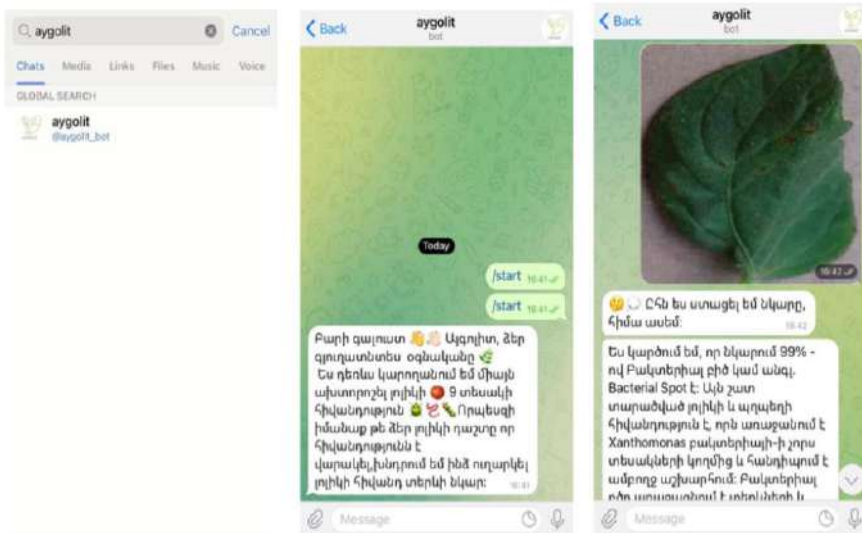
def help(update, context):
    update.message.reply_text("""
/start - Սկսել խորհրդատվությունը
/help - Ցույց է տալիս այս տեքստը
""")
```

Նկ. 5. Telegram Bot-ում օգտատերերին ավտոմատ կեպով պատասխանելու ֆունկցիայի օրինակ:

Փուլ 7. «aygolit» Telegram Bot-ը գործարկելու համար անհարժեշտ քայլերը նկարագրված են նկար 6-ում: Ծրագիրը փորձարկելու համար տվյալների բազայի այն մասից, որը դեռ մեր մոդելը չէր տեսել, ընտրեցինք «բակտերի-

ալ բիծ» հիվանդությամբ վարակված լոլիկի տերևի նկար, այն Telegram Bot-ի միջոցով մեր ծրագրին ուղարկելուց հետո ստացանք նկար 6-ի քայլ 3-ի պատասխանը: Ընդհանուր դեպքում մեր մշակած ծրագիրը ունի 94% ճշտություն, այսինքն՝ դեպքերի 94 տոկոս դեպքում այն ճիշտ է ակստորոշում:

Նկ. 6. Ծրագրի վերջնական տեսքը:



Քայլ 1

Քայլ 2

Քայլ 3

Սույն հոդվածը ներկայացնում էր խորը ուսուցման մասնավոր ճյուղի պատկերների դասակարգման կիրառությունը ագրարային ոլորտում: Ստացված արդյունքները վստահեցնող են, առաջիկայում կարելի է մեծացնել տվյալների բազան և, ինչու ոչ, ավելացնել նոր բույսերի հիվանդությունների տվյալներ:

Օգտագործված գրականության ցանկ

1. <https://www.expert.ai/blog/machine-learning-definition/>
2. <https://www.altexsoft.com/blog/image-recognition-neural-networks-use-cases/>
3. Classifying CIFAR10 images using ResNets, Regularization and Data Augmentation in PyTorch <https://jovian.ai/aakashns/05b-cifar10-resnet>

-
4. Pytorch documentation <https://pytorch.org>
 5. Aurelien Geron (2019) “Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition.”
 6. Atharva Ingle GitHub code can be found here <https://github.com/Gladiator07>.

APPLICATION OF DEEP LEARNING IN CROP PRODUCTION

ABGARYAN HENRIK

Faculty of Agribusiness and Economics

Bachelor Student

e-mail: henri.abgaryan@gmail.com

MANUCHARYAN DONARA

Chair of Data Science, ANAU

Lecturer

e-mail: mdon2008@yandex.ru

Deep learning (DL) is an artificial intelligence application that allows computers to learn and improve. Deep learning, with its incredible potential to calculate and analyze huge amounts of data, is an advanced tool for completing complex tasks faster and more efficiently. The main focus of deep learning is computer programs that can easily obtain the necessary data and use them for training. Data can be statistical analyses, personal data and other similar information from any area and in any size. Deep learning has many applications.

This article is about one of the most important components of deep learning such as image classification. The article presents a brief history of deep learning, various models of the image classification; in particular, neural networks and their practical application in the agricultural sector, including detailed descriptions and graphs, program codes in language Python.

Keywords: artificial intelligence, machine learning, neural networks, database, Kaggle, Python, Google Colabrotory, package network (Convolutional Neural Network), TensorFlow.

ПРИМЕНЕНИЕ ГЛУБОКОГО ОБУЧЕНИЯ В РАСТЕНИЕВОДСТВЕ

АБГАРЯН ГЕНРИК

Факультет агробизнеса и экономики НАУА,

бакалавр

электронная почта: henri.abgaryan@gmail.com

МАНУЧАРЯН ДОНАРА

Преподаватель кафедры датологии НАУА

электронная почта: mdon2008@yandex.ru

Глубокое обучение (ГО) является применением искусственного интеллекта, которое позволяет компьютерам учиться и совершенствоваться. Глубокое обучение с его невероятным потенциалом для расчета и анализа огромных объемов данных является передовым инструментом для более быстрого и эффективного выполнения сложных задач. Основным инструментом глубокого обучения являются компьютерные программы, которые могут легко получить необходимые данные и использовать их для обучения. В качестве данных могут использоваться как статистические, так и другие подобные показатели из любой области в любом количестве. Глубокое обучение имеет множество применений.

Статья знакомит с машинным обучением и одним из его важнейших компонентов - классификацией изображений. В статье представлена краткая история машинного обучения, различные модели классификации изображений, в частности нейронные сети, практическое применение в аграрной сфере, включая подробные описания, графики, программный код на языке Python.

Ключевые слова: искусственный интеллект, машинное обучение, нейронные сети, база данных, Kaggle, Python, Google Colabrotory, пакетная сеть (Convolutional Neural Network), TensorFlow.

Հոդվածը ներկայացվել է խմբագրական խորհուրդ 25.05.2022թ.:

Հոդվածը գրախոսվել է 01.06.2022թ.: