INSTITUTE OF COMPUTER SCIENCES
Master in Artificial Intelligence and Data Science

Universitätsstr. 1 D–40225 Düsseldorf

Heinrich Heine Universität Düsseldorf

AI-based fluorescent labeling for cell line development

Hanna Pankova

Master thesis

Date of issue: 01. April 2022
Date of submission: 29. August 2022
Reviewers: Prof. Dr. Markus Kollmann
Dr. Wolfgang Halter

Erklärung

Hiermit versichere ich, dass ich diese Master thesis selbstständig verfasst habe. Ich habe dazu keine anderen als die angegebenen Quellen und Hilfsmittel verwendet.

Düsseldorf, den 29. August 2022

Hanna Pankova

Abstract

Cell line development is an expensive and time-consuming process, however that is the most modern approach for producing the proteins needed in pharmaceuticals.

Contents

1 Introduction

1.1 Motivation

1.2 Notation

2 Background

2.1 Biology

Cell line development is a process of generating single cell-derived clones that produce high and consistent levels of target therapeutic protein. (pharma.lonza.com/offerings/mammalian/cell-line-development)

Differential interference contrast (DIC) microscopy is an optical microscopy technique used to enhance the contrast in unstained, transparent samples (wikipedia).

Proteins are large biomolecules and macromolecules that comprise one or more long chains of amino acid residues (https://en.wikipedia.org/wiki/Protein).

Fluorescent labelling is the process of covalently binding fluorescent dyes to biomolecules such as nucleic acids or proteins so that they can be visualized by fluorescence imaging (https://www.nature.com/subjects/fluorescent-labelling). A fluorophore is a chemical compound that can reemit light at a certain wavelength.These compounds are a critical tool in biology because they allow experimentalists toimage particular components of a given cell in detail. (O'reilly life sciences p113)

Cell line development (CLD) is the process by which the cellular machinery is co-opted to manufacture therapeutic biologics or other proteins of interest. One can use different expression systems for cell line development: bacterial, plant-based, yeast, mammalian. (copy paste from https://www.beckman.de/resources/product-applications/lead-optimization/cell-line-development) Chinese hamster ovary (CHO) cells are the most popular mammalian cells used for protein production. (doi:10.1016/B978-0-08-100623-8.00007-4)

First step of CLD is the introduction of the gene of interest (GOI or a DNA vector) to CHO cells. This process is called a transfection. It is important to transfect a GOI into an optimal site of genome to secure a high protein expression over time during protein production, however pratically transfection mostly results in a vector being inserted into a random site within a host cell genome. In case the gene was transfected in the inactive site of genome (and the majority of genome is not transriptionally active) the cell will likely not express the gene. (doi:10.1016/B978-0-08-100623-8.00007-4) (doi:10.1016/j.coche.2018.08.002)

The second step is the selection of cell pools that have successful and stable gene integrations. The reasond why not all of them are suitable for cloning is the following: during the transfection only 80% of cells recieve the vector of GOI (doi:10.1016/B978-0-08-100623-8.00007-4), only the small percent of which actually integrate a vector into the genome and, as mentioned above, only a fraction of those cells are able to stabily express a protein. (Reference needed). Such selection could be done with bulk sorting algorithm. (doi:10.1016/B978-0-08-100623-8.00007-4)

The third step in CLD is to clone the cells. The chosen stable pools of cells are phenotypically and genetically diverse - meaning they have different growth rates, metabolic profile and etc. This is not ideal for industrial production - all the cells used for protein production should be derived from the same clone ([25] here

doi:10.1016/B978-0-08-100623-8.00007-4). In order to choose single best cells for further cloning one asseses several parameters like cell size, granularity, cell surface protein expression and etc. This can be done with Fluorescent Activated Cell Sorting (FACS) technology. (https://doi.org/10.1517/14712598.4.11.1821). Unfortunately fluorescence labeling is expensive and may ruin the cell due to its phototoxicity (https://doi.org/10.1371/journal.pone.0007497). There is a limited number of available fluorescent channels in microscopes as well as such labels can also be inconsistent, depend a lot on reagent quality, and require many hours of lab work. Therefore there exists a need for flurescent labeling in silico - without intervening into the cell.

Once the cells are cloned, phenotypical and genetical heterogeneity is reduced, the next step is to charaterize clonally-derived cells based on the following criteria: cell size, growth rate, protein quality, titer, metabolities and etc. With this one can estimate clones productivity and titer. Such observations may take up to 90 days after which one can determine which cells are stable and therefore suitable for production. This is the last step of CLD process and consumes a lot of time and maintaining costs for feeding and cloning the cells. Predicting the stability of the cells directly from DIC images would reduce this time significantly allowing to escape this process completely.

However there are also some disadtantages of this approach. First, it can be less accurate than skilled cells staining perfomed manually. Extreme or unsual clones and phenotypes might be challenging if they were not used in the training set of images.

2.2 Imaging

The microscope used in the experiments takes photos of the well plate in random locations. The reason for that hides in the focusing problem, to get a reasonably good photo without blur it has to focus on a specific location, this is done there automatically, therefore the location of the focus is almost random. Although it might be problematic in the following sense: photos takes by the microscope in such manner do not gurantee that the focus will land in distinct spots all the time. This means that some cells taken during one of the photos might appear in the later ones. Since the photos have a high-resolution the crops are first performed and it might happen that same cell might appear in several crops. Afterwards, when crops are split between train, test and validation datasets it might happen that the same cell will once land in the train set and another time in the validation set, which will lead to a not completely fair and representative validaiton loss during training.

2.3 ML

Background on Unet and ML in general

Convoutional neural network is a neural network that is based on convolutional layers. It is a powerful tool for image processing and is used in medical imaging. Convolution is a linear operation used in convolutional layers that can be performed by applying a kernel (a 2d matrix) across a bigger input matrix called tensor, which can be 3d. Elementwise product between them is calculated and summed, this value will be an element of

the output 2d matrix. Kernel slides accross all locations of the input tensor. In case if several different kernels were used then a 3d tensor will be created.

Main advantage of convolutional neural networks if weight sharing. Kernel has learnable weights however these weights are shared across all locations of the kernel on the input tensor, thic strongly reduces the number of parameters needed.

CNNs also uses non-linearities like RELU, ELU, Tahn, Sigmoids and etc. They are also often combined with max pooling layers and dropouts to escape overfitting.

Overfitting is one of the most often problems in deep learning that prevents model to generalize well for unseen data. This can happen when the model is too big for the amount of training data given, it was not regularized well or there is just not enough data for training.

U-Net architecture is widely used for segmentation purposes. It is a convolutional neural network with the following architecture: [img] . It first performs image downsampling and upsampling afterwards. [https://arxiv.org/pdf/1505.04597.pdf] The following architecture have been used for nuclei prediction.

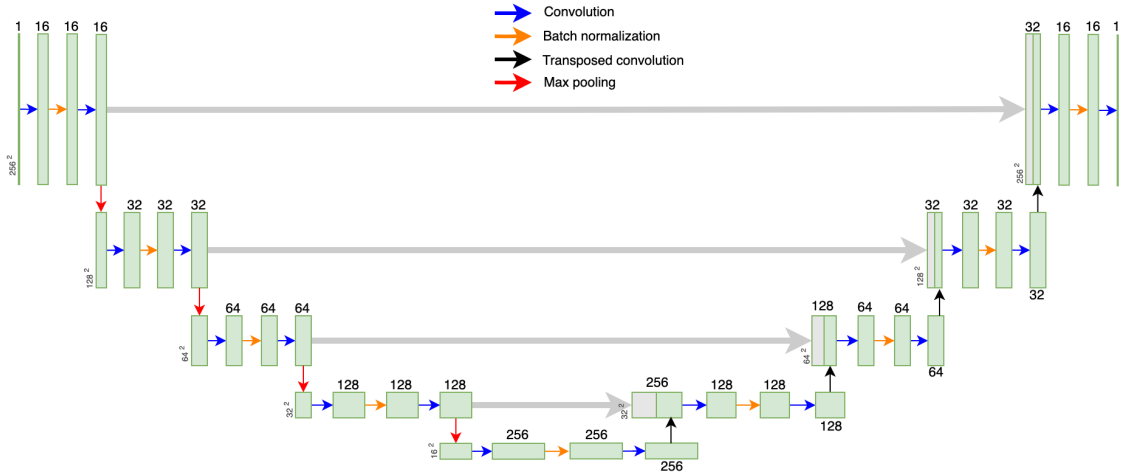


Figure 1: Unet

3 Prediction of cell organelles

3.1 Overfitting

Definition 3.1 (Overfitting)**.** "Hypothesis overfits the training samples if some other hypothesis that fits the training samples less well actually performs better over the entire distribution of instances" (p67 Mitchell Machine Learning 1997).

Overfiting prevents model to generalize well on the unseen data and in order to avoid fitting to closely to the training dataset one has several options:

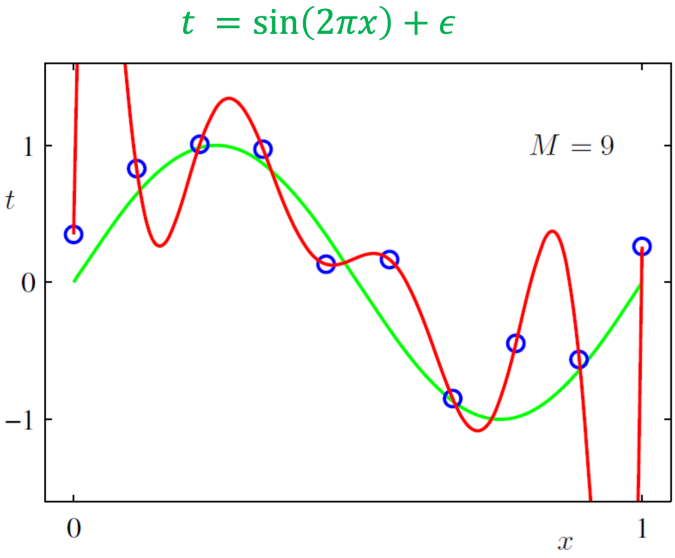


Figure 2: Overfitting

(Bishop book)

3.1.1 Early-stopping

Overfitting effect occurs at later epochs. This doesn't happen suring early epochs as with the correct weights initialization the weights of the model are quite small and random and therefore the best decision surface would be a smooth one. But in the later epochs the difference in values of the weights grows and they become not similar anymore which means also that the decision surface becomes more complex and will be ble to fit not only the training data itself, but also its noise. (p111 Mitchell Machine Learning 1997). And that is why stopping before the model became too complex for the given data may mitigate this problem.

3.1.2 Regularization

The complexity of the model grows with the number of features it uses, sometimes the model may pay attention to the features that are not important to the outcome, or even considers a noise to be a feature. To prevent this one should decrease the weights associated with useless features, however we cannot know ahead which of them should be ignored, therefor one limits them all. (doi:10.1088/1742-6596/1168/2/022022) For that one adds a penalty term in loss function:

$$\tilde{L}(\theta, X, y) = L(\theta, X, y) + \lambda R(\theta) \tag{1}$$

for some $\lambda > 0$. This is call a *soft-constrait* optimization. When $R(\theta)$ is of form $R(\theta) = ||\theta||_2^2 = \sqrt{\sum_i \theta_i^2}$ this is called *L2-regularization* and when it is of form $R(\theta) = |\theta||_1 = \sum_i |\theta_i|$ this is called *L1-regularization*. *L2-regularization* used with backpropagation is equivalent to weight decay. Weight decay is defined by Hanson and Pratt (1988) as following:

$$\theta_{t+1} = (1 - \lambda)\theta_t - \alpha \frac{\partial L}{\partial \theta_t} \tag{2}$$

where α is a learning rate. Weight decay successfully affect more those weights the gradient change along which is smaller (Goodfellow Deep learning p229). *L1-regularization* induces sparsity of the weights by assining some of them to zero, this could be also considered as feature selection approach.

Regularization techniques like BatchNorm and Dropout could also be applied. BatchNorm is defined by :

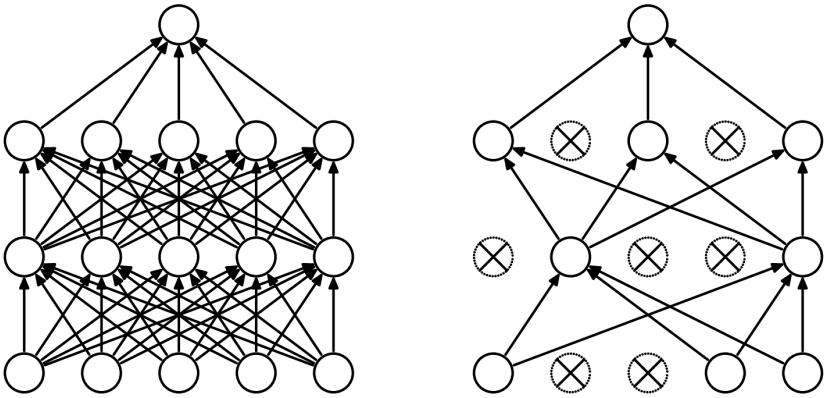
$$\begin{aligned} y_i &= \gamma \frac{x_i - \mu_B}{\sigma_B^2 + \epsilon} + \beta \\ \sigma_B^2 &= \frac{1}{m} \sum_i^m (x_i - \mu_B)^2 \\ \mu_B &= \frac{1}{m} \sum_i^m x_i \end{aligned} \tag{3}$$

where m is a batch size. Ioffe and Szegedy, 2015

Dropout is a technique that randomly sets some of the weights to zero. (Srivastava, Hinton 2014).

Early stopping in combination with weight decay, BatchNorm were used to regularize the model that was overfitting too quickly with Dropout only. This could be done either by reducing the number of the parameters, therefore by changing the model archticture. Or by adding regularization layers like BatchNorm or Dropout. Regularization can also be done by putting more resctritions on weights by adding their $L1$ or $L2$ norms to the loss function.

From the first training one can clearly see an overfit happening around epoch 30. Although one could just pick out one of epochs before the 30th epoch before overfit has



(a) Standard Neural Net (b) After applying dropout.

Figure 3: Dropout

happened, to apply more regularizion to the model that has been using dropout only would be a good idea. Early stopping in combination with weight decay, BatchNorm were used to regularize the model that was overfitting too quickly with dropout only. $BatchNorm$ layers have been added after the first Convolution layer in each ConvBlock and $TransposedConvBlock$. The results of training the regularized network is presented in Figure 5.

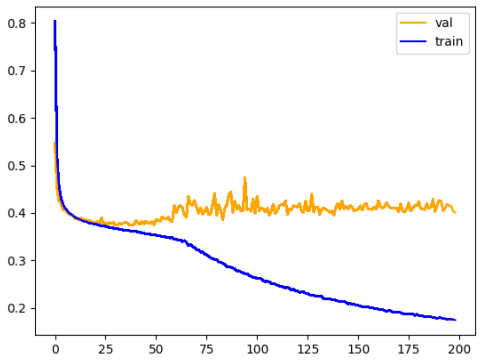


Figure 4: Not regularized

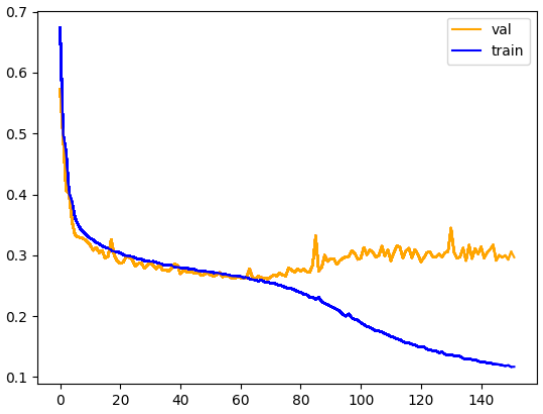


Figure 5: Regularized

4 Summary

List of Figures

List of Tables