IOWA STATE UNIVERSITY
Department of Animal Science

An introduction to machine learning and the other AI

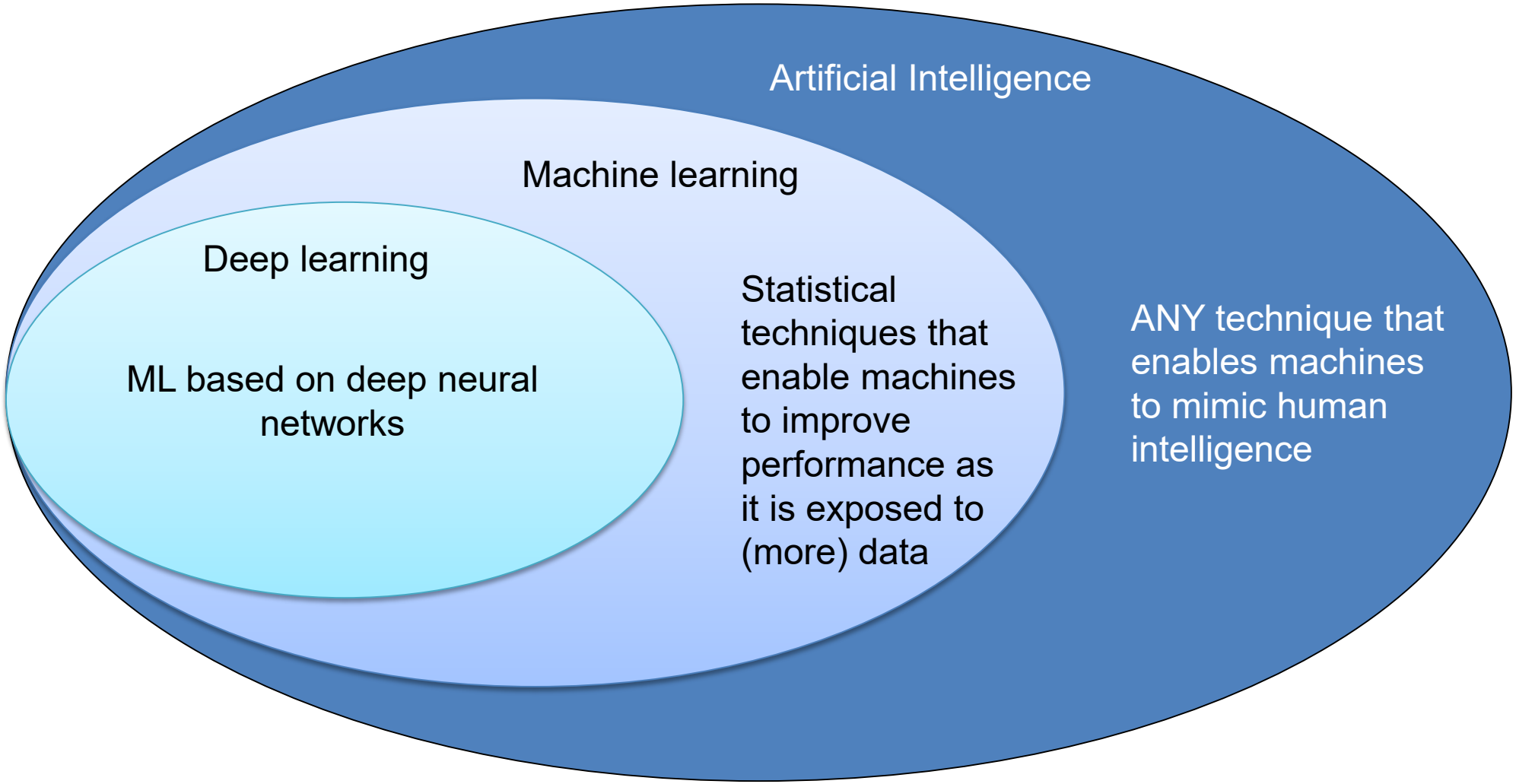
Potential applications to beef cattle

Juan P. Steibel

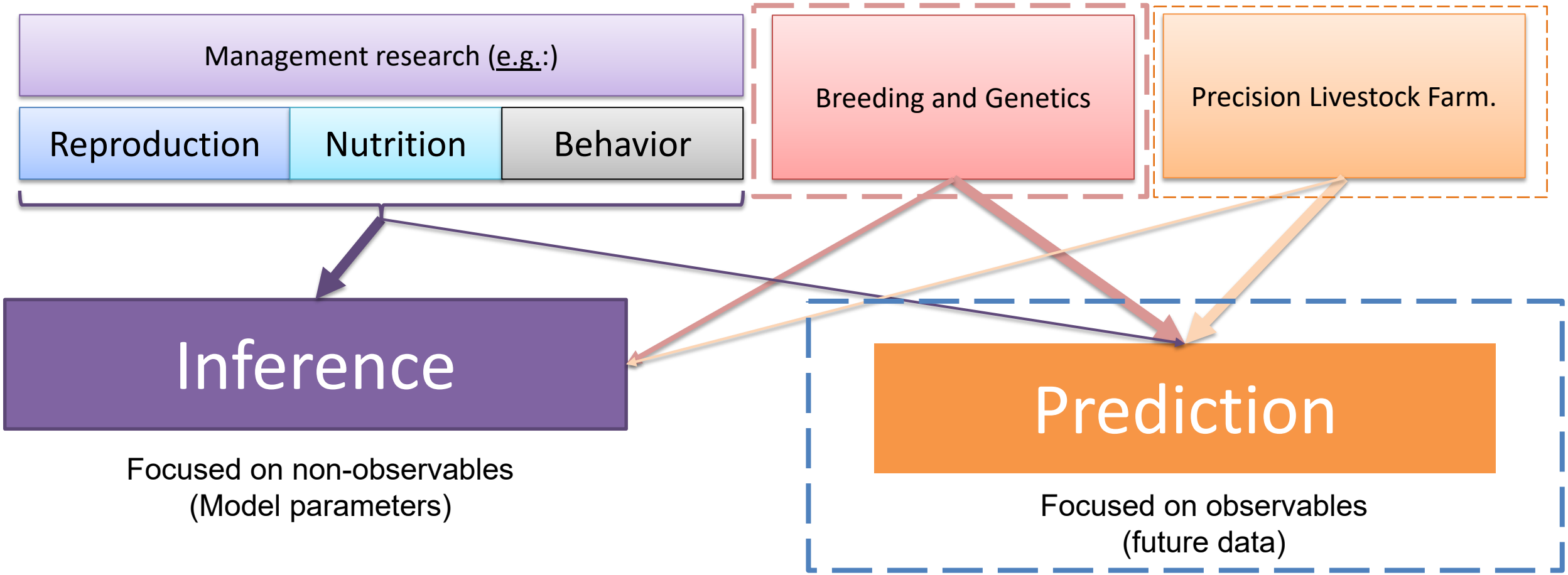
Professor of Animal Science

J. Lush Endowed Chair of Animal Breeding and Genetics

Definitions



Statistics applied to animal science, animal breeding and PLF



Data Science / Statistics Tasks compared

	Prediction	Inference
General Goal:	Predict new outcomes	Learn about data generation process
Specific Target:	Prediction accuracy	High power, low FDR
Focused on:	Observables (future data)	Non-observables (parameters)
Model Checking:	Cross validation	Goodness-of-fit
Model Uncertainty:	Model averaging	Model selection
Data source:	Observational data	Designed experiments
Example:	Predict breeding values and phenotypes	Find candidate variants, genes, pathways

Challenge in prediction for animal breeders: Increase prediction accuracies

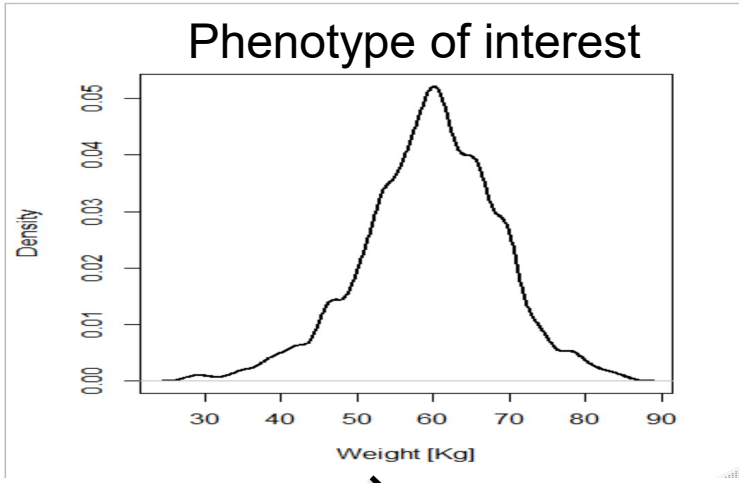
- More and better data

Genomics	Phenomics
Increasing marker density	Novel phenotypes
Better annotation?	Traditional phenotypes in more animals
More individuals	More conditions/environments/timepoints
Multi-population (breeds, etc) panels	Multi-layer omics
More environmental data (GxE)	More and better metadata

- Alternative model (-ing)?

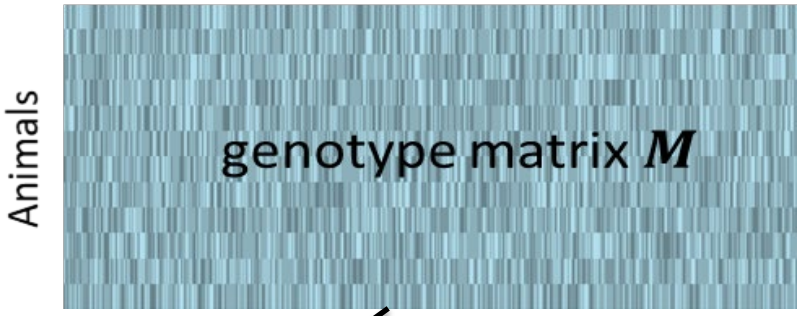
Application: Genomic Prediction

Phenotype of interest



Non-genetic covariates (metadata)

SNP markers



$$Y = Xb + Mg + e$$

Penalty (regularization) + selection

Genomic Breeding Value:

$$\hat{a} = m_i \hat{g}$$

Meuwissen et al 2001

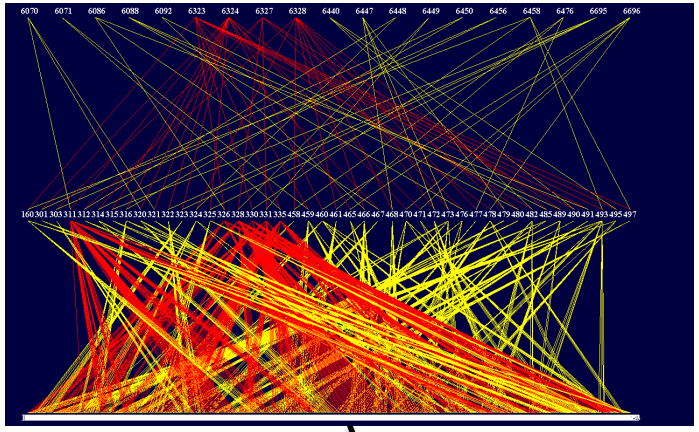
Application: GBLUP

Marker-centric model

$$\boldsymbol{Y} = \boldsymbol{Xb} + \boldsymbol{Mg} + \boldsymbol{e}$$

$$\boldsymbol{g} \sim N(\boldsymbol{0}, \boldsymbol{I}\sigma_a^2)$$

Pedigree



$$\boldsymbol{a}^* \sim N(\boldsymbol{0}, \boldsymbol{A}\sigma_a^2)$$

Lush, Hazel 1940s
Henderson 1950s-1970s

Animal-centric model

$$\boldsymbol{Y} = \boldsymbol{Xb} + \boldsymbol{a} + \boldsymbol{e}$$

$$\boldsymbol{a} \sim N(\boldsymbol{0}, \boldsymbol{G}\sigma_a^2)$$

$$\boldsymbol{G} = \boldsymbol{MM}'$$

Van Raden 2007
Stranden and Garrick 2009

"Single-step": Combines these two
Misztal et al 2000s

Application: GBLUP

$$\boldsymbol{Y} = \boldsymbol{Xb} + \boldsymbol{Mg} + \boldsymbol{e}$$

$$\boldsymbol{g} \sim \boldsymbol{N}(\boldsymbol{0}, \boldsymbol{I}\boldsymbol{\sigma}_{\boldsymbol{a}}^{\boldsymbol{2}})$$

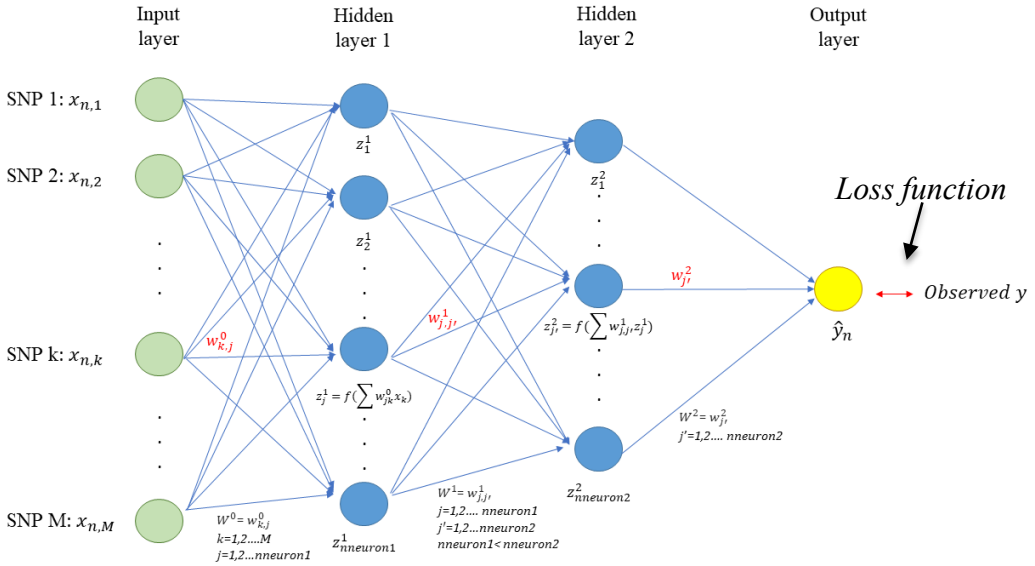
- ✓ Still, by far, the most used model for genomic prediction
- ✓ Distributional assumptions can be relaxed (Bayesian Alphabet: Fernando, Goddard, etc)
- ✓ Linear model
- ✓ Genetic Additive model (No dominance, no epistasis)
- ✓ Independent SNP contributions (Tempelman et al.: antedependence priors)

Deep learning in genomic prediction.

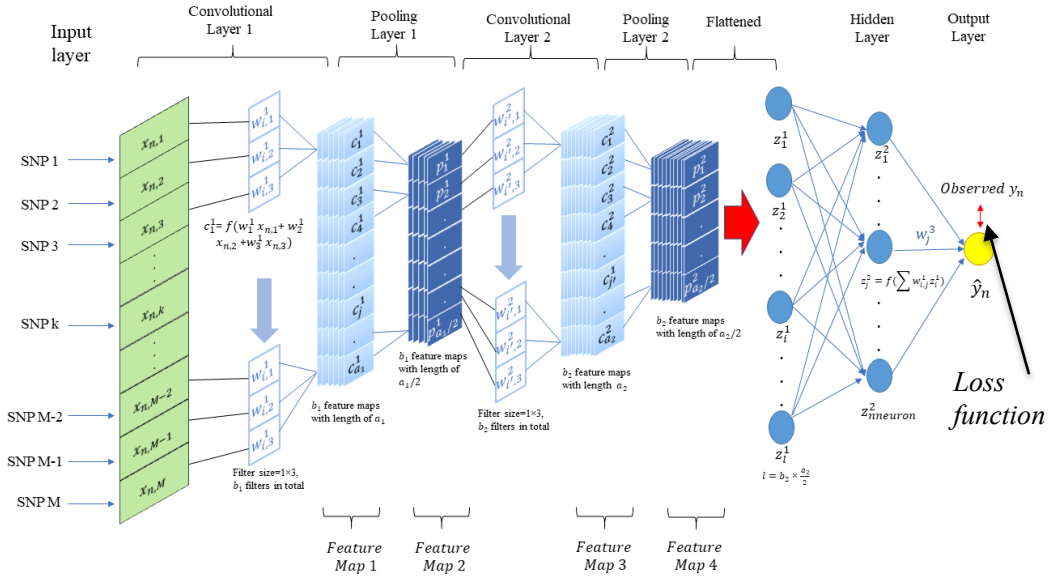
- Nonlinear prediction
- Models interactions between features
- CNN account for spatial patterns (LD)
- It should work better than GBLUP, right?

Junjie Han
Github

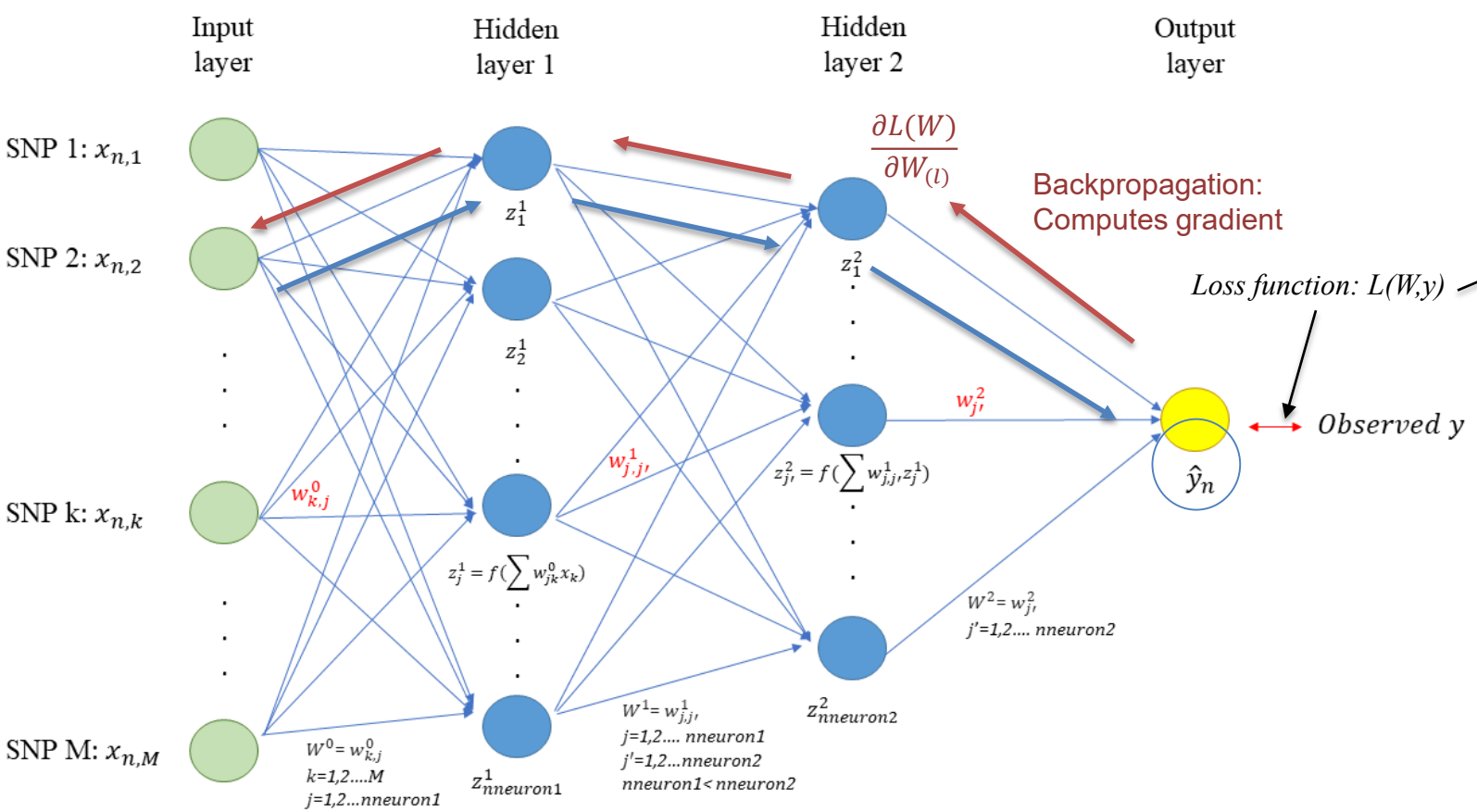
MLP: Multilayer perceptron

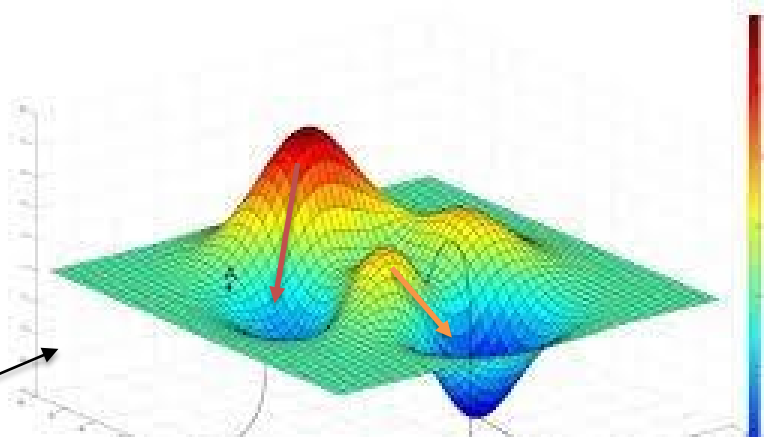


CNN: Convolutional Neural Network



Training (fitting) Deep Learning models





Results sensitive to:

- ✓ Network architecture
- ✓ Constraints (regularization, dropout)
- ✓ Batch size (subset of observations used in back propagation + optimization)
- ✓ Optimizer and learning rate
- ✓ Number of epochs (iterations)

(Some) hyperparameters in deep learning

Table 1 – Parameter space for optimized hyperparameters

Hyperparameters	Parameter space (MLP)	Parameter space (CNN)	Value Type
Number of layers	[1,2,3,4,5]	[1,2,3,4,5]	Integer
Number of neurons	[8-512]	[8-512]	Integer
Activation	['relu', 'elu', 'sigmoid', 'selu', 'softplus', 'linear', 'tanh']	['relu', 'elu', 'sigmoid', 'selu', 'softplus', 'linear', 'tanh']	Categorical
Optimizer	['sgd', 'adam', 'adagrad', 'rmsprop', 'adadelta', 'adamax', 'nadam']	['sgd', 'adam', 'adagrad', 'rmsprop', 'adadelta', 'adamax', 'nadam']	Categorical
Dropout rate	[0-1]	[0-1]	Continuous
L2 penalty	[0-1]	[0-1]	Continuous
Batch size	[N×α_1-N×α_2]	32	Integer
Epoch	[21-50]	[21-50]	Integer
Number of filters	NA	[2-128]	Integer
Filter size	NA	[2-20]	Integer
Pooling	NA	['max', 'average']	Categorical

Hyperparameter space and range (see details in File S1). N represents sample size. α_1=0.001 for the simulated datasets and α_1=0.01 for the real pig dataset. α_2=0.01 for the simulated datasets and α_2=0.1 for the real pig dataset.

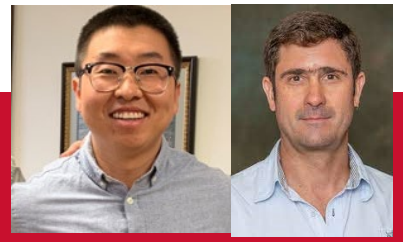
Structural properties
Optimizer properties
Effects constraints

It's all about the tradeoff between overfitting and underfitting ☺

Heuristic hyperparameter optimization of deep learning models for genomic prediction

Junjie Han, Cedric Gondro, Kenneth Reid, Juan P Steibel ✉

G3 Genes|Genomes|Genetics, Volume 11, Issue 7, July 2021, jkab032,
https://doi.org/10.1093/g3journal/jkab032



What hyperparameter combinations were selected?

- ✓ Many alternative models performed equally well
- ✓ Real data required more complex models
  - ✓ More layers
  - ✓ Nonlinear activation functions
- ✓ More complex models required stronger regularization

Alternative datasets

Table 3 – Hyperparameters of selected MLP models from each population

Dataset	DE No.	Activation	No. layer(s)	No. neurons	Batch	Epoch	Optimizer	Dropout	L2
SP	1	elu	2	[446,87]	51	37	adam	0.006	0.06
SP	2	elu	2	[412,150]	41	45	adam	0.020	0.16
SP	3	elu	2	[470,155]	46	44	adam	0.015	0.06
SP	4	selu	2	[474,145]	54	45	adam	0.032	0.13
SP	5	softplus	2	[397,87]	54	45	adam	0	0.13
SC	1	elu	3	[429,330,57]	44	28	adam	0.030	0.04
SC	2	relu	2	[411,106]	48	41	adamax	0.002	0.06
SC	3	elu	3	[401,269,93]	11	27	adamax	0.001	0.01
SC	4	relu	1	409	56	21	adam	0.034	0.14
SC	5	relu	1	444	47	33	adam	0.020	0.16
RP	1	sigmoid	3	[374,192,25]	10	40	sgd	0.352	0.85
RP	2	sigmoid	3	[476,193,69]	54	42	adam	0.480	0.52
RP	3	sigmoid	3	[483,291,8]	44	46	adamax	0.182	0.12
RP	4	sigmoid	3	[457,234,79]	31	41	adamax	0.465	0.03
RP	5	sigmoid	3	[386,251,148]	8	40	sgd	0.617	0.75

SP, simulated pig dataset; SC, simulated cattle dataset; RP, real pig dataset; DE No., differential evolution of different data partition; No. layer(s), number of hidden layers; No. neurons, number of neurons according to the number of hidden layers.

Alternative datasets

Table 4 Hyperparameters of selected CNN models from each population

Data	DE No.	Activation	No. layers	No. filters	Filter size	Epoch	FCL	Optimizer	Dropout	L2	Pooling
SP	1	linear	1	110	19	25	17	Adamax	0.197	0.21	Average
SP	2	Elu	1	16	15	32	110	rmsprop	0.146	0.03	Average
SP	3	Elu	1	15	8	44	79	rmsprop	0.692	0.02	Average
SP	4	linear	1	59	20	24	49	adamax	0.496	0.23	max
SP	5	linear	1	109	13	27	109	adam	0.827	0.01	Average
SC	1	linear	1	116	20	30	16	adam	0.370	0.10	Average
SC	2	linear	1	87	12	25	12	adam	0.086	0.13	Average
SC	3	linear	1	32	8	42	24	adam	0.250	0.19	Average
SC	4	linear	1	79	20	44	27	adamax	0.666	0.06	Max
SC	5	linear	1	98	16	40	153	adam	0.151	0.17	Average
RP	1	elu	2	[51,113]	18	22	50	adam	0.277	0.67	Average
RP	2	relu	3	[24,81,121]	12	27	268	adam	0.067	0.11	Average
RP	3	elu	2	[64,112]	13	45	278	adam	0.021	0.87	Average
RP	4	relu	3	[44,73,106]	13	47	326	adam	0.008	0.18	Average
RP	5	elu	3	[41,71,128]	5	41	238	adam	0.051	0.35	Average

Deep learning in genomic prediction.

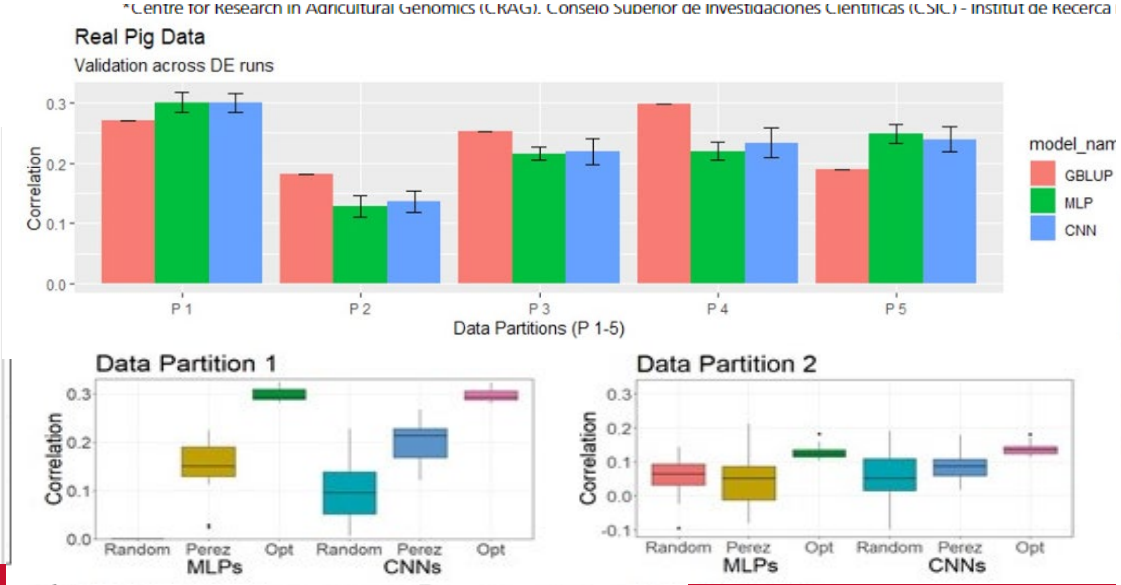
- Nonlinear prediction
- Models interactions between features
- CNN account for spatial patterns (LD)
- It should work better than GBLUP, right?

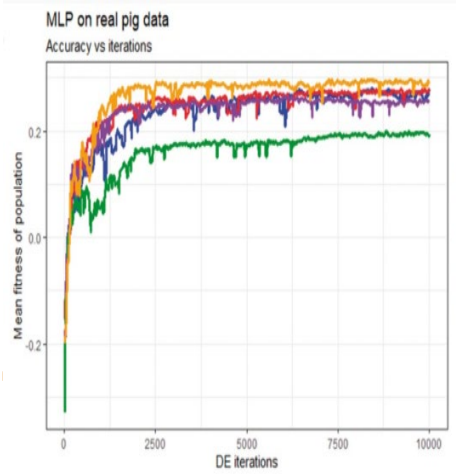
Heuristic hyperparameter optimization of deep learning models for genomic prediction

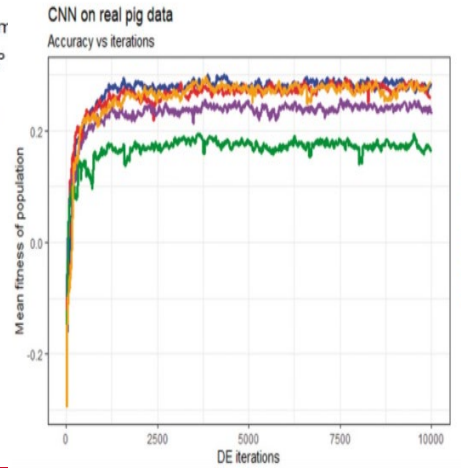
Junjie Han, Cedric Gondro, Kenneth Reid, Juan P Steibel

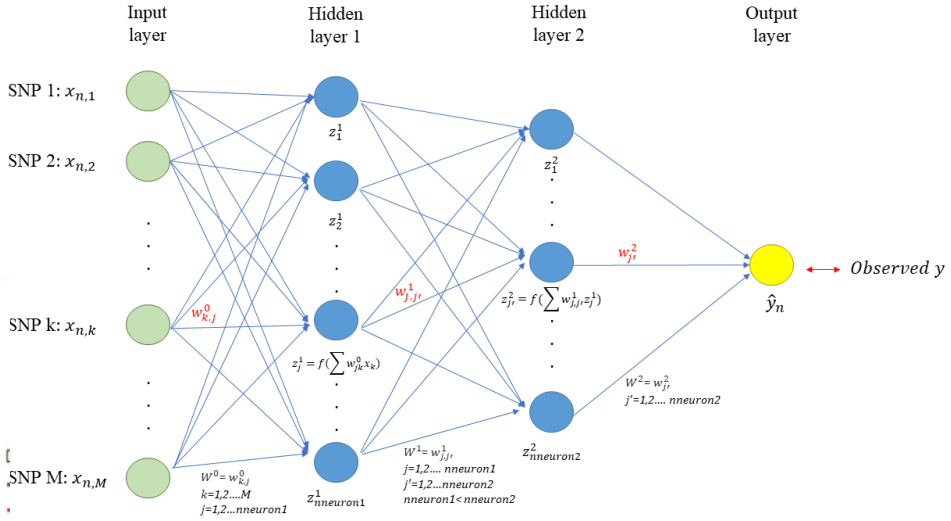
G3 Genes|Genomes|Genetics, Volume 11, Issue 7, July 2021, jkab032,
https://doi.org/10.1093/g3journal/jkab032

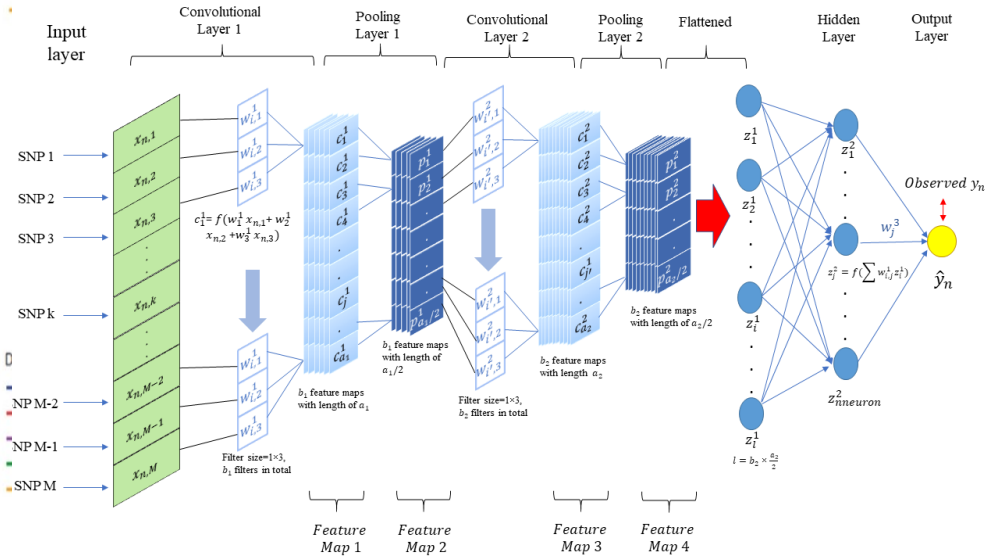
Pig data, N=910, 29K SNP, Phenotype= Ph-24 hrs









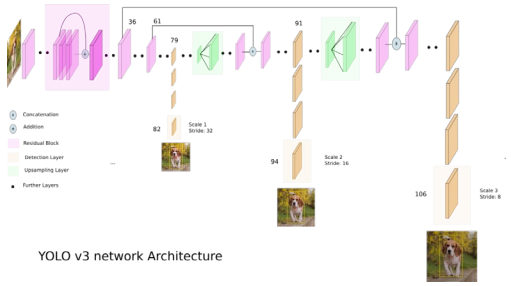


Why deep learning work(ed) in other applications?
There is no GBLUP for CV 8^D

Object detection

COCO Dataset

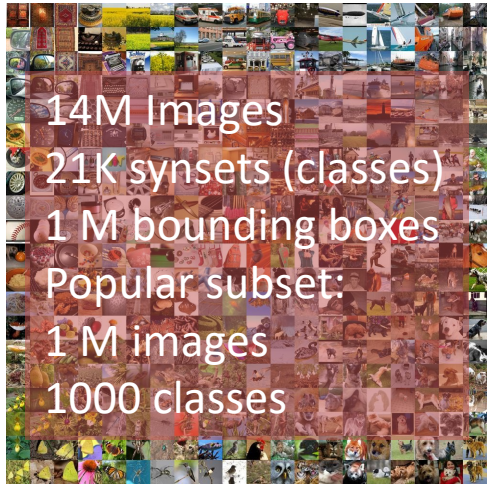
YOLO



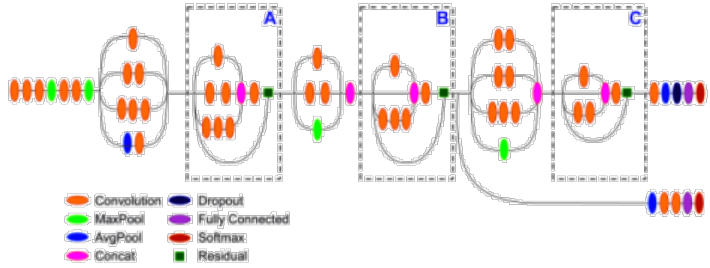
YOLO v3 network Architecture

Image classification

ImageNet Dataset
CIFAR-10/100 Dataset

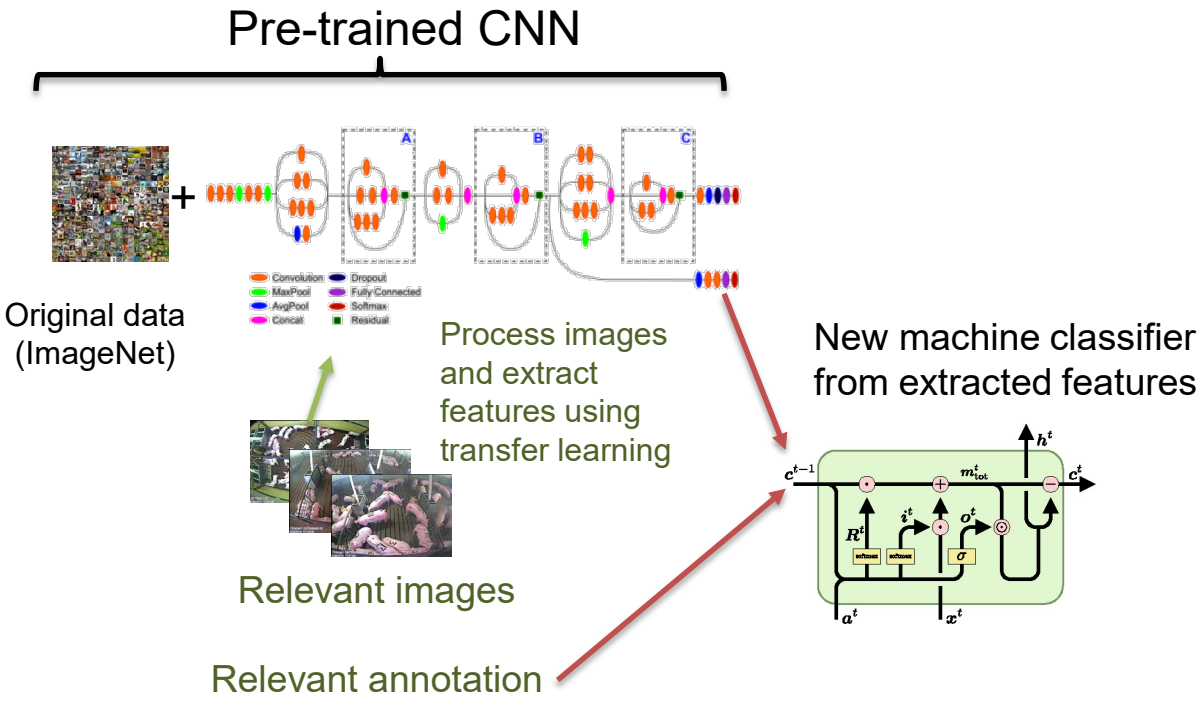


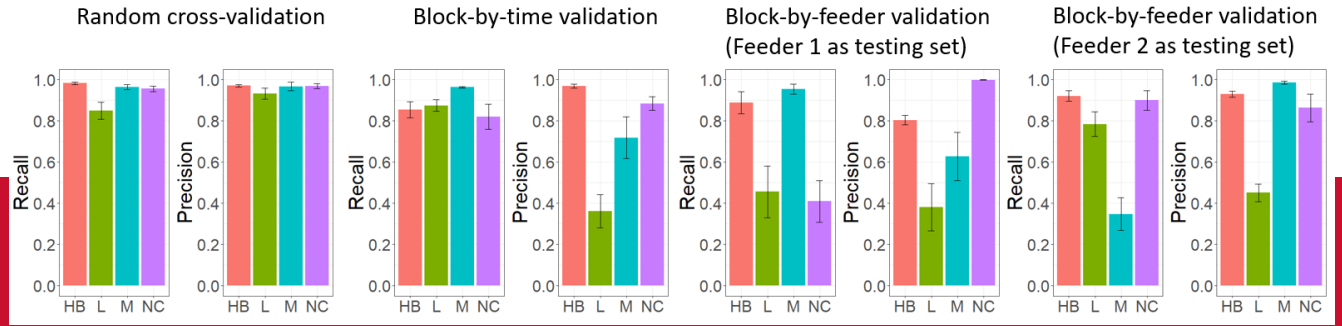
Inception-vx
ResNet50
VGG-x
etc



How it is being used in CV-based phenotyping ?

In general, a combination of transfer learning re-training is used with a **proven DL architecture**

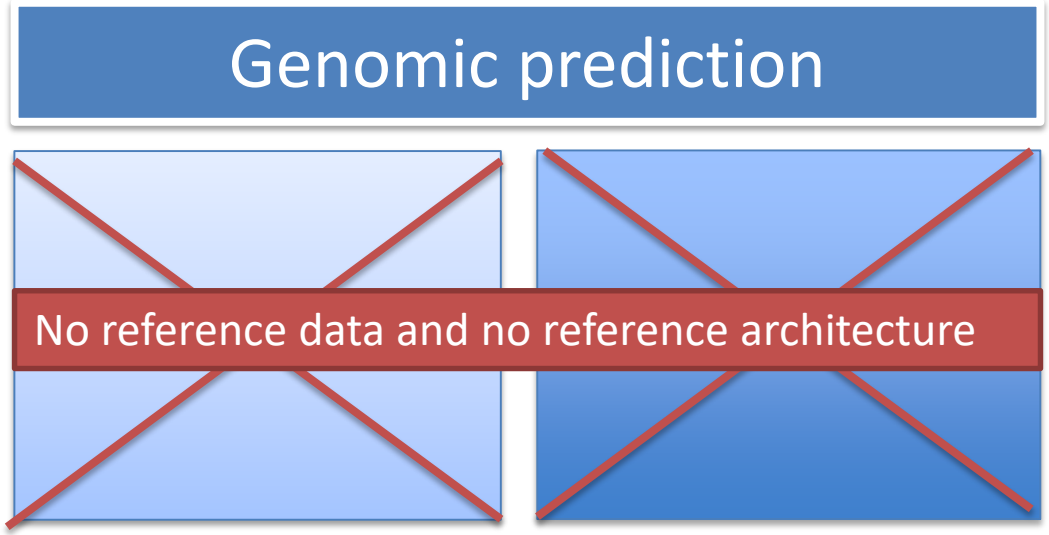




High performance is usually obtained

Validation is the key:
Most studies use internal cross-validation

Revisit DL and genomic prediction?

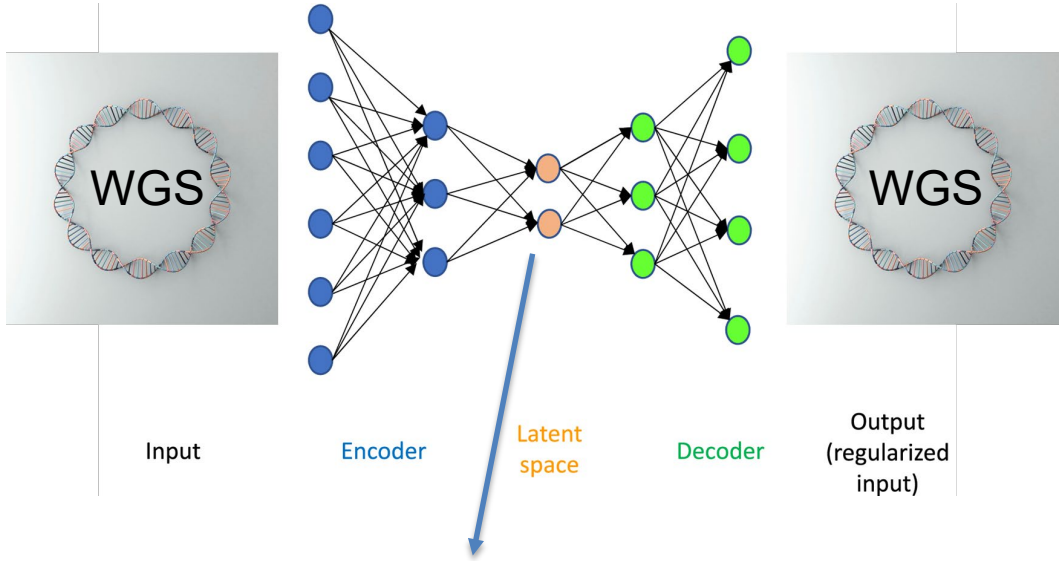


Do we need this?

If we will keep trying to use DL for Genomic prediction… Maybe.

Think of more challenging phenotypes (e.g: predicting shapes)

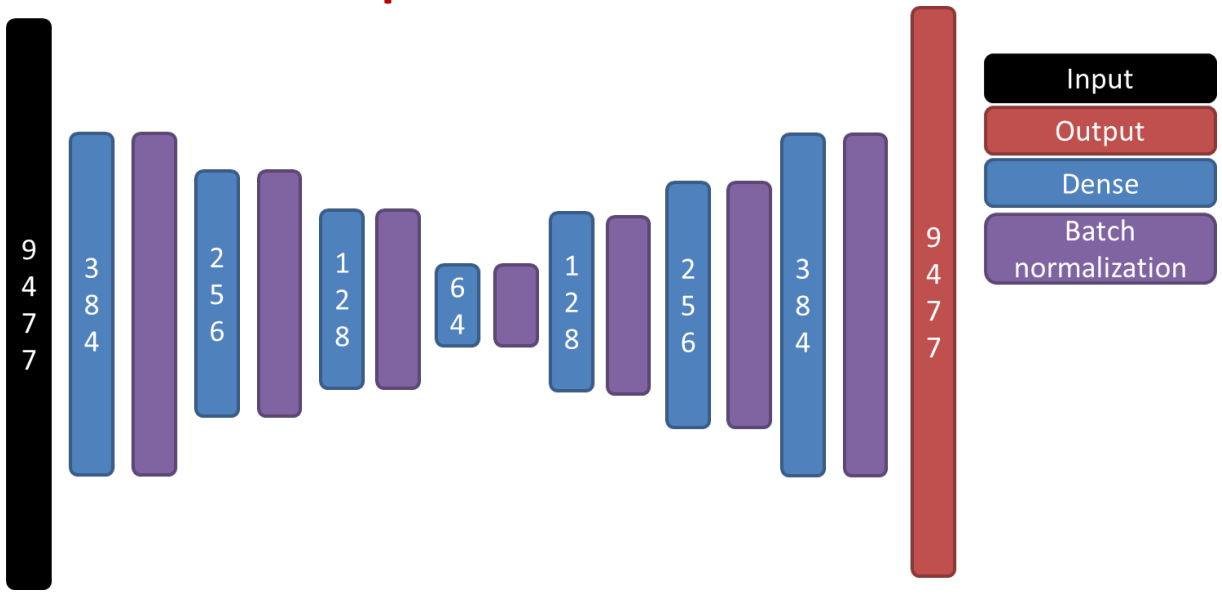
Idea: Use autoencoders to obtain a reduced representation of livestock genomes



Use this latent space in genomic prediction

Other uses of autoencoders:
- Anomaly detection (breed purity)
- Denoising (imputation)
- puzzle solvers (assembly?)

Naïve Example: autoencoder for anomaly detection (breed purity)



- M=9477 SNP markers, N=2000 Duroc pigs, 1795 training, 4-fold cross validation.
- Testing: 2000 synthetic individuals from duroc+yorkshire genotypes
- Pre-processing: Filter low call rate, Naïve imputation of missing
- Hyperparameters: Epocs=<=200 with early stopping, Batch size 5, Optimizer adagrad, Activation elu
- Evaluation Criteria: Cor(observed,reconstructed)

Quantiles of the distribution of correlations between reconstructed and observed genotype

	0.01	0.025	0.975	0.99
Purebred Duroc – Autoencoder	0.66	0.686	0.85	0.86
Purebred Duroc – PCA	0.30	0.33	0.76	0.80
“Crossbred” - Autoencoder	0.26	0.27	0.44	0.45

Advantage of this approach: single breed reference panel used

Conclusion

- Animal Breeders have ample experience with prediction problems
  - We are used to interpret something out of our black box models
  - With genomic predictions we became experts in cross validation
- Machine Learning has proven very useful in MULTIPLE applications
  - Deep learning (a type of ML) is particularly useful in computer vision.
  - This algorithms are black box at its best
  - Predictive performance over cross validation is all that matters
- If we want to apply DL/ML for BV/phenotypic prediction, we need to get a better grasp of genome variation
- If we want to apply DL for phenomics (e.g: computer vision) we need larger and more heterogeneous datasets.