Intellectual Property on AI

Expert-Driven Protections,
Business-Driven Results

WIPO Conversation on Intellectual Property and Frontier Technologies - AI Inventions

Daeho Lee, Managing Partner

PI IP LAW
LEE, PARK & ASSOCIATES

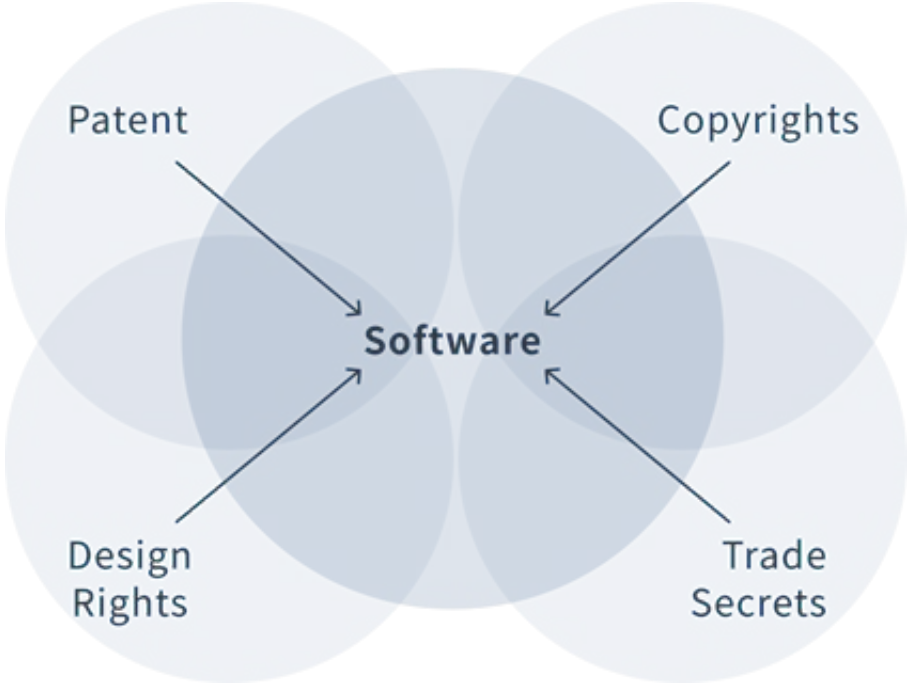
PI IP LAW

Frequently asked questions

COPYRIGHTS	Q. Copyrights cover the protection of software. Why do we also need a patent?
OPEN INNOVATION	Q. Why do we need patent protection on AI even though innovation on AI technologies benefits from open innovation, such as open source software policy?
ENFORCEMENT	Q. How can software patent owners collect evidence for infringement?

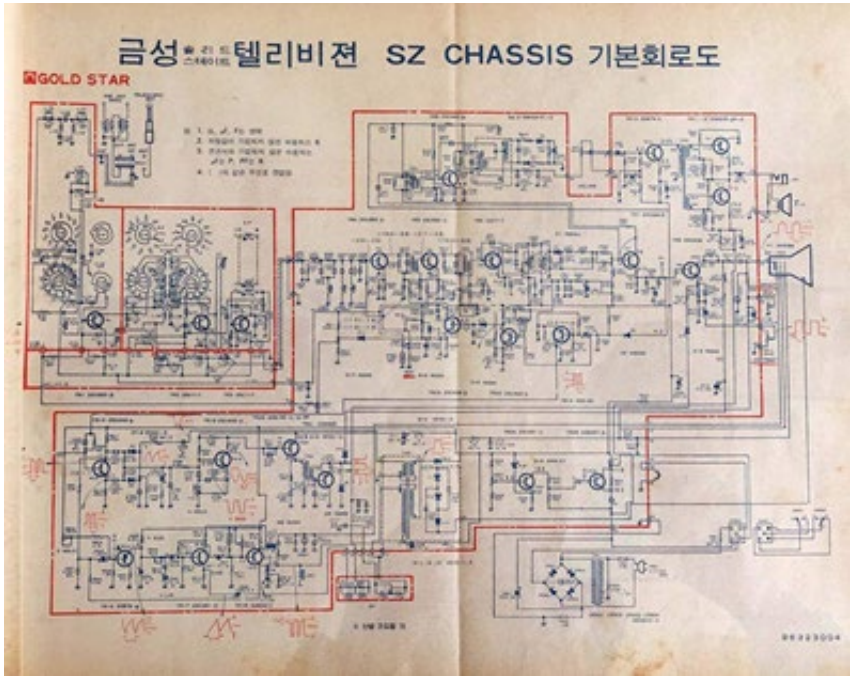
PI IP LAW

Aspects of the protection



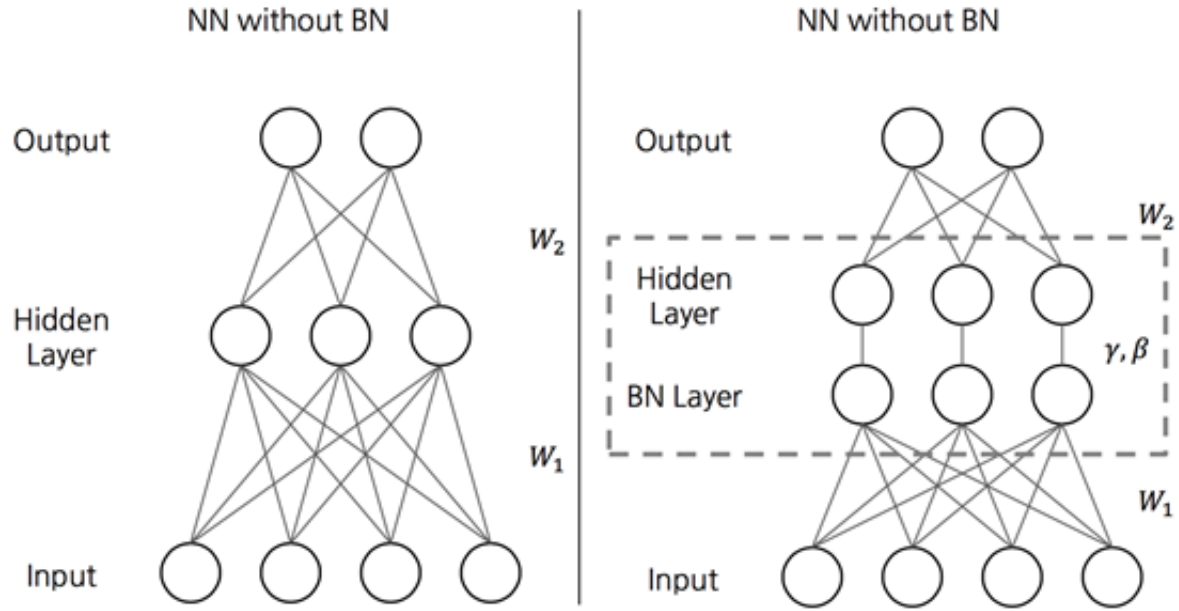
PI IP LAW

Open Innovation



PI IP LAW

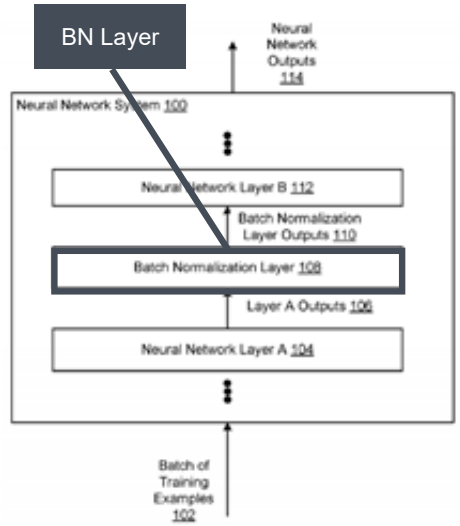
Open Innovation



Patents over technologies included in Open Source Software

BATCH NORMALIZATION LAYERS (GOOGLE, 2016)

Representative Figure



Claim

1. A neural network system implemented by one or more computers, the neural network system comprising:

(1) a batch normalization layer between a first neural network layer and a second neural network layer, wherein the first neural network layer generates first layer outputs having a plurality of components, and wherein the batch normalization layer is configured to, during training of the neural network system on a batch of training examples:

receive a respective first layer output for each training example in the batch;

compute a plurality of normalization statistics for the batch from the first layer outputs;

(1) normalize each component of each first layer output using the normalization statistics to generate a respective normalized layer output for each training example in the batch;

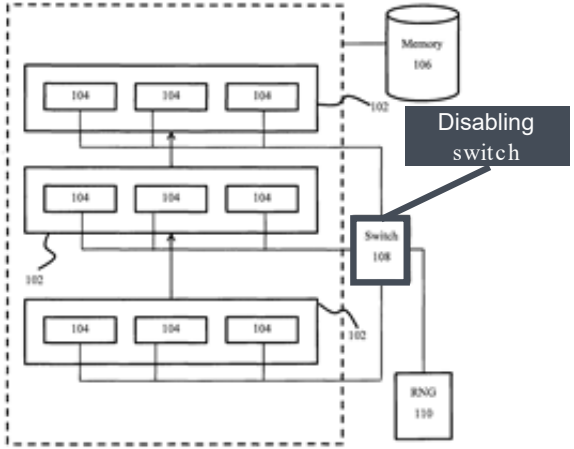
generate a respective batch normalization layer output for each of the training examples from the normalized layer outputs; and

provide the batch normalization layer output as an input to the second neural network layer.

Patents over technologies included in Open Source Software

SYSTEM AND METHOD FOR ADDRESSING OVERFITTING IN A NEURAL NETWORK (GOOGLE, 2016)

Representative Figure



Claim

1. A computer-implemented method comprising:

obtaining a plurality of training cases; and

training a neural network having a plurality of layers on the plurality of training cases, each of the layers including one or more feature detectors, each of the feature detectors having a corresponding set of weights, and a subset of the feature detectors being associated with respective probabilities of being disabled during processing of each of the training cases, wherein training the neural network on the plurality of training cases comprises, for each of the training cases respectively:

determining one or more feature detectors to disable during processing of the training case, comprising determining whether to disable each of the feature detectors in the subset based on the respective probability associated with the feature detector,

❶disabling the one or more feature detectors in accordance with the determining, and

processing the training case using the neural network with the one or more feature detectors disabled to generate a predicted output for the training case.

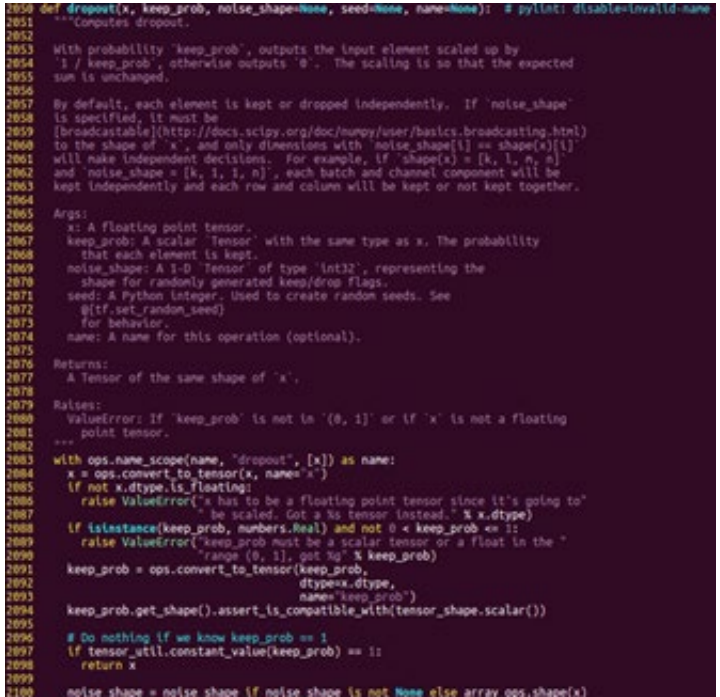
PI IP LAW

Patents over technologies included in Open Source Software

Although Google open-sourced TensorFlow and provides many features as libraries, such as Batchnorm, Dropout, etc., Google also registered patents over the technologies.

Google can not enforce these patents on the users of TensorFlow. However, it still maintains exclusive rights to the above technologies outside the TensorFlow community.

Dropout Library implemented in TensorFlow

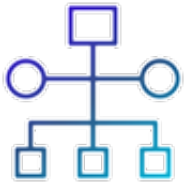


```
def dropout(x, keep_prob, noise_shape=None, seed=None, name=None):  # pylint: disable=invalid-name
  """Computes dropout.

  With probability `keep_prob`, outputs the input element scaled up by
  `1 / keep_prob`, otherwise outputs `0`.  The scaling is so that the expected
  sum is unchanged.

  By default, each element is kept or dropped independently.  If `noise_shape`
  is specified, it must be
  [broadcastable](http://docs.scipy.org/doc/numpy/user/basics.broadcasting.html)
  to the shape of `x`, and only dimensions with `noise_shape[i] == shape(x)[i]`
  will make independent decisions.  For example, if `shape(x) = [k, l, m, n]`
  and `noise_shape = [k, 1, 1, n]`, each batch and channel component will be
  kept independently and each row and column will be kept or not kept together.

  Args:
    x: A floating point tensor.
    keep_prob: A scalar `Tensor` with the same type as x. The probability
      that each element is kept.
    noise_shape: A 1-D `Tensor` of type `int32`, representing the
      shape for randomly generated keep/drop flags.
    seed: A Python integer. Used to create random seeds. See
      @{tf.set_random_seed}
      for behavior.
    name: A name for this operation (optional).

  Returns:
    A Tensor of the same shape of `x`.

  Raises:
    ValueError: If `keep_prob` is not in `(0, 1]` or if `x` is not a floating
      point tensor.
  """
  with ops.name_scope(name, "dropout", [x]) as name:
    x = ops.convert_to_tensor(x, name="x")
    if not x.dtype.is_floating:
      raise ValueError("x has to be a floating point tensor since it's going to"
                       " be scaled. Got a %s tensor instead." % x.dtype)
    if isinstance(keep_prob, numbers.Real) and not 0 < keep_prob <= 1:
      raise ValueError("keep_prob must be a scalar tensor or a float in the "
                       "range (0, 1], got %g" % keep_prob)
    keep_prob = ops.convert_to_tensor(keep_prob,
                                      dtype=x.dtype,
                                      name="keep_prob")
    keep_prob.get_shape().assert_is_compatible_with(tensor_shape.scalar())

    # Do nothing if we know keep_prob == 1
    if tensor_util.constant_value(keep_prob) == 1:
      return x

    noise_shape = noise_shape if noise_shape is not None else array_ops.shape(x)
```

PI IP LAW

Types of Claims available for AI patent

Computer- programs



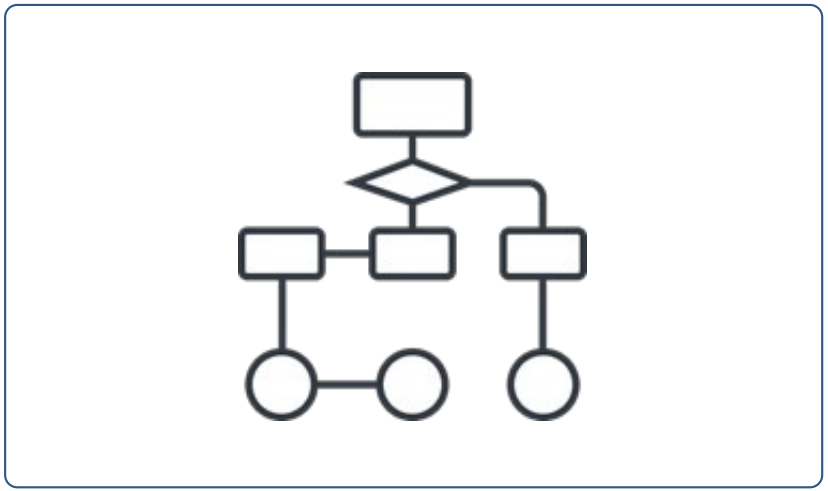
Methods

Devices

Traditional types of claims available for software patents are also available for AI patents

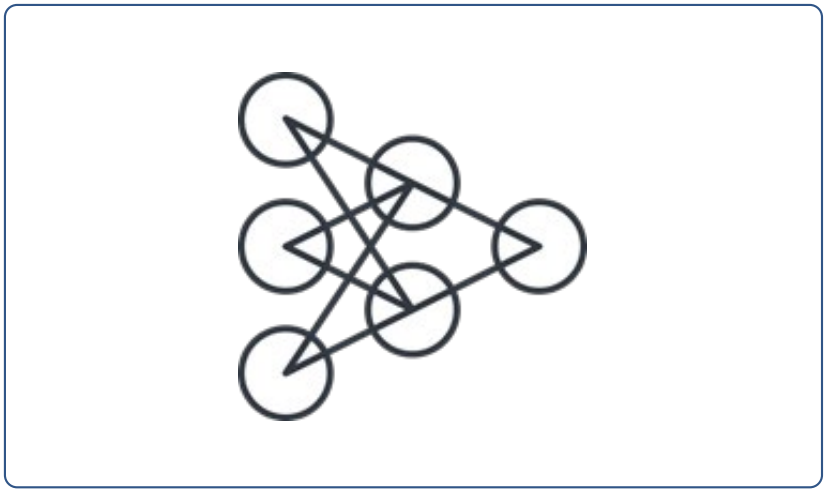
PI IP LAW

How to draft AI Claims



Rule-based Software

The rule-based software performs the predetermined process set by the algorithms. The methods claims can describe the steps of the algorithms.

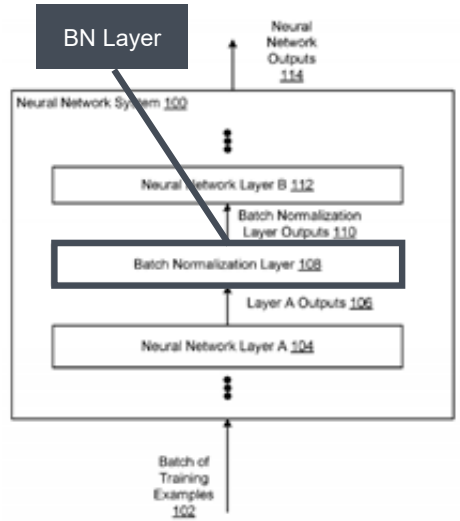


Data-Driven AI models

Deep learning models do not follow the fixed process. The models perform inference on the given problem based on data used for training.

How to draft AI Claims

Representative Figure



Claim

1. A neural network system implemented by one or more computers, the neural network system comprising:

(1) a batch normalization layer between a first neural network layer and a second neural network layer, wherein the first neural network layer generates first layer outputs having a plurality of components, and wherein the batch normalization layer is configured to, during training of the neural network system on a batch of training examples:

receive a respective first layer output for each training example in the batch;

compute a plurality of normalization statistics for the batch from the first layer outputs;

normalize each component of each first layer output using the normalization statistics to generate a respective normalized layer output for each training example in the batch;

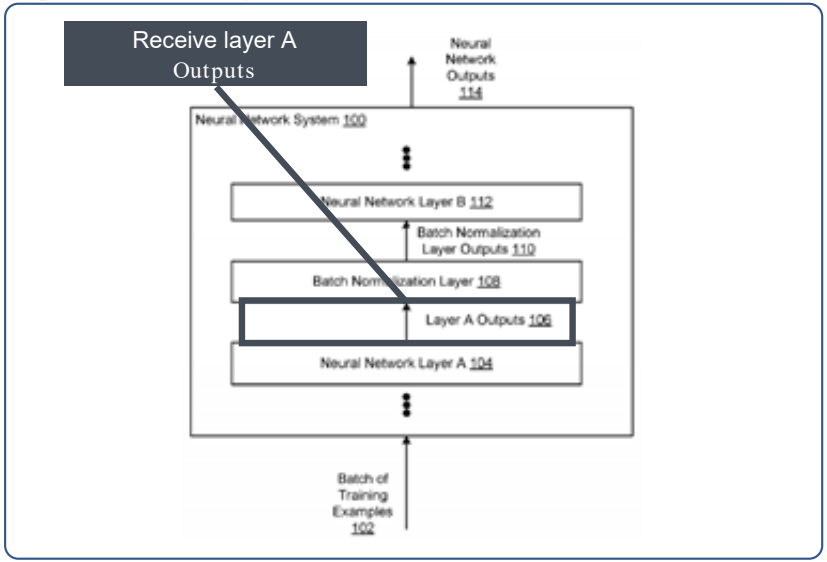
generate a respective batch normalization layer output for each of the training examples from the normalized layer outputs; and

provide the batch normalization layer output as an input to the second neural network layer.

The description of a BN LAyer

How to draft AI Claims

Representative Figure



Claim

1. A neural network system implemented by one or more computers, the neural network system comprising:

a batch normalization layer between a first neural network layer and a second neural network layer, wherein the first neural network layer generates first layer outputs having a plurality of components, and wherein the batch normalization layer is configured to, during training of the neural network system on a batch of training examples:

(2) receive a respective first layer output for each training example in the batch;

compute a plurality of normalization statistics for the batch from the first layer outputs;

normalize each component of each first layer output using the normalization statistics to generate a respective normalized layer output for each training example in the batch;

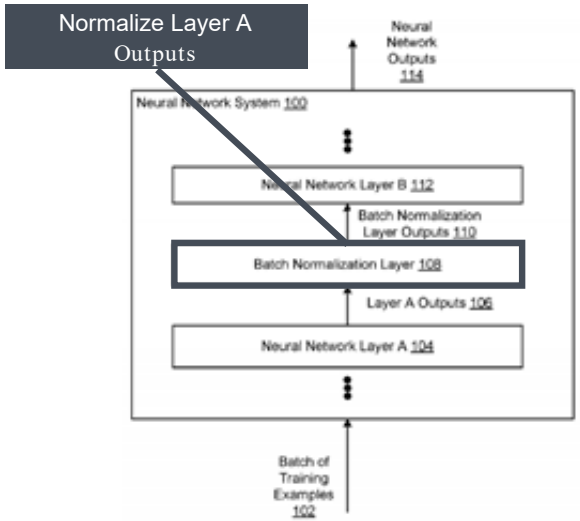
generate a respective batch normalization layer output for each of the training examples from the normalized layer outputs; and

provide the batch normalization layer output as an input to the second neural network layer.

Data (Layer A Outputs) flows from Layer A to the BN Layer.

How to draft AI Claims

Representative Figure



Claim

1. A neural network system implemented by one or more computers, the neural network system comprising:

a batch normalization layer between a first neural network layer and a second neural network layer, wherein the first neural network layer generates first layer outputs having a plurality of components, and wherein the batch normalization layer is configured to, during training of the neural network system on a batch of training examples:

receive a respective first layer output for each training example in the batch;

3 compute a plurality of normalization statistics for the batch from the first layer outputs;

normalize each component of each first layer output using the normalization statistics to generate a respective normalized layer output for each training example in the batch;

generate a respective batch normalization layer output for each of the training examples from the normalized layer outputs; and

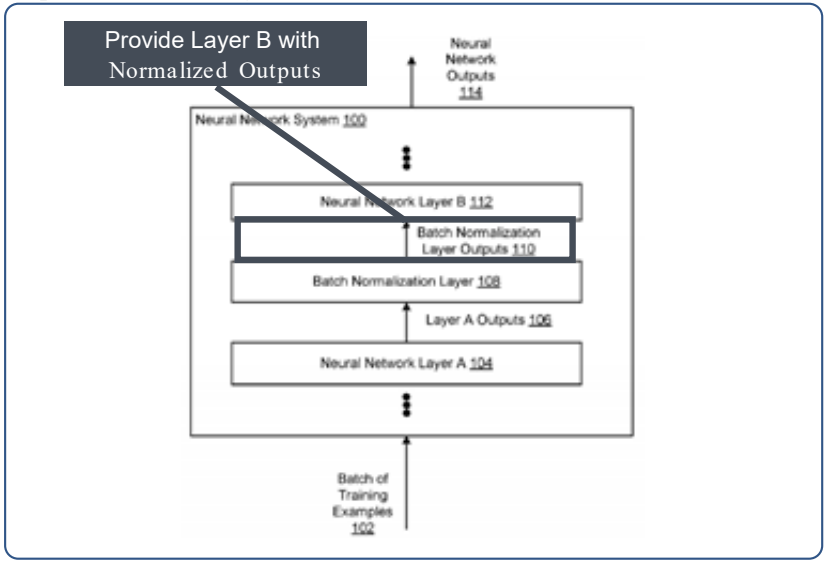
provide the batch normalization layer output as an input to the second neural network layer.

The BN Layer computes the norm stats (StdDev. and Avg.) and normalizes the Layer A outputs with the computed norm stats.

PI IP LAW

How to draft AI Claims

Representative Figure



Claim

1. A neural network system implemented by one or more computers, the neural network system comprising:

a batch normalization layer between a first neural network layer and a second neural network layer, wherein the first neural network layer generates first layer outputs having a plurality of components, and wherein the batch normalization layer is configured to, during training of the neural network system on a batch of training examples:

receive a respective first layer output for each training example in the batch;

compute a plurality of normalization statistics for the batch from the first layer outputs;

normalize each component of each first layer output using the normalization statistics to generate a respective normalized layer output for each training example in the batch;

generate a respective batch normalization layer output for each of the training examples from the normalized layer outputs; and

4 provide the batch normalization layer output as an input to the second neural network layer.

Data (BN Layer Outputs flows from the BN Layer to Layer B.

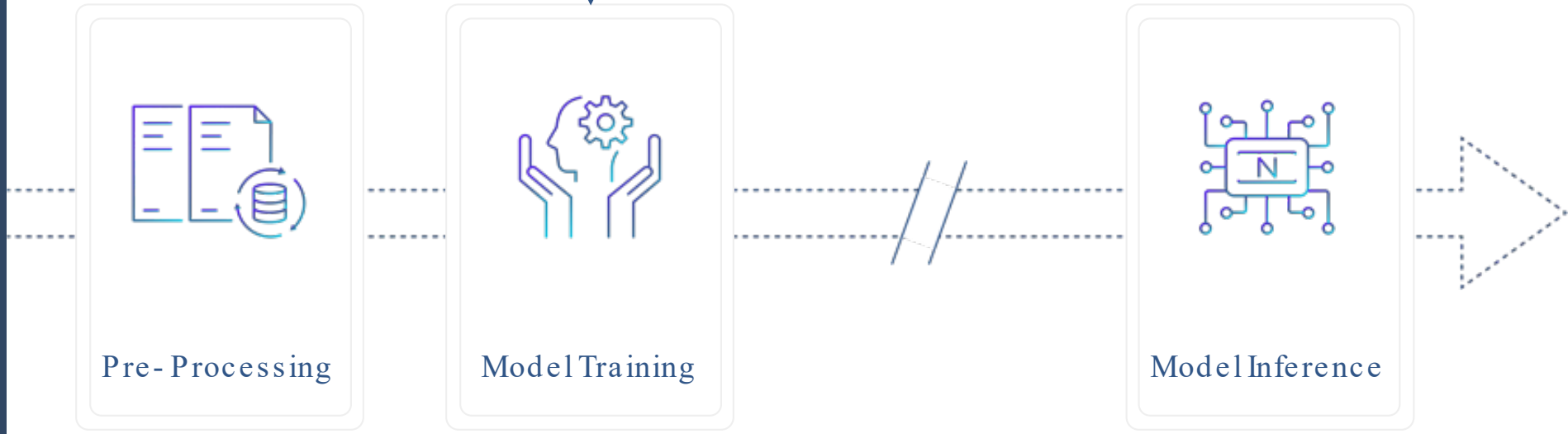
PI IP LAW

How to draft AI Claims

PI IP LAW

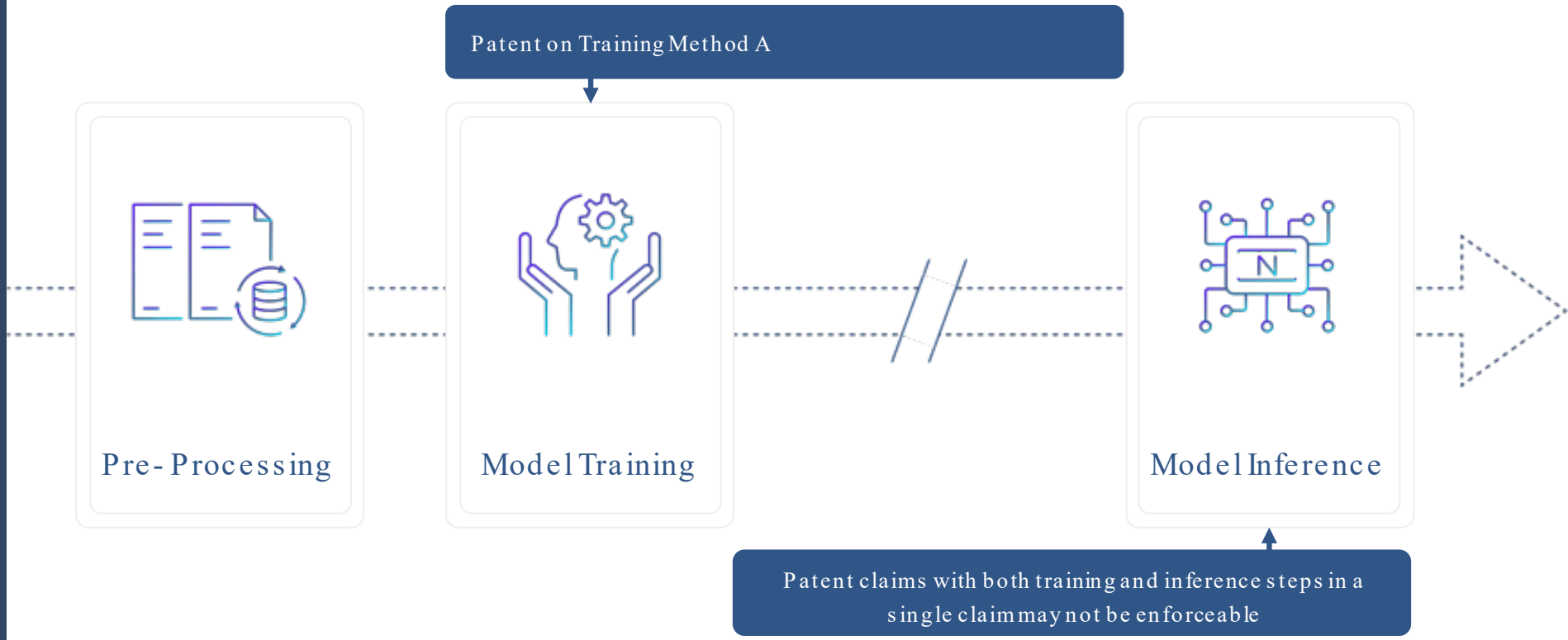
How to draft AI Claims

Patent on Training Method A



PI IP LAW

How to draft AI Claims



Checklists for AI Inventions

PI IP LAW

STRICTLY CONFIDENTIAL

Pre-Processing

Check Point 01

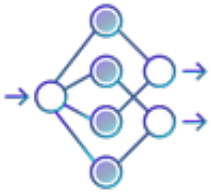
Any unique process to collect/augment/clean training data used for machine learning?

Training

Check Point 02

Any additional module used only for the training process or unique data flow different from the ordinary training method?

Model



Check Point 03

Any unique points on AI model structure and design?

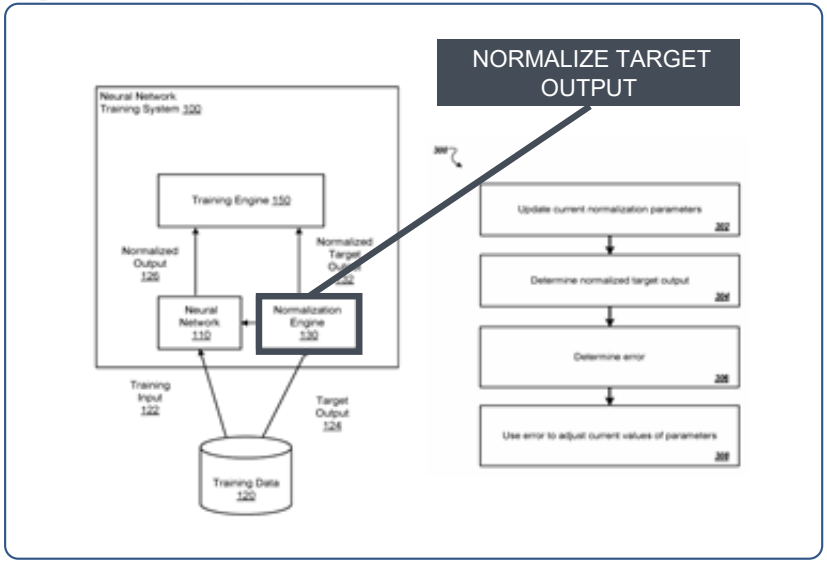
Post Processing

Check Point 04

Any unique process using output of AI model to perform desired goal of the invention?

Training Method Claims

Representative Figure



Claim

1. A method for training a neural network on training data to generate normalized outputs that are mappable to un-normalized outputs in accordance with a set of normalization parameters, wherein the training data comprises a sequence of training items and, for each training item in the sequence, a respective target output, the method comprising, for each training item in the sequence:

updating current values of the normalization parameters to account for the target output for the training item;

(1) determining a normalized target output for the training item by normalizing the target output for the training item in accordance with the updated normalization parameter values;

processing the training item using the neural network to generate a normalized output for the training item in accordance with current values of main parameters of the neural network;

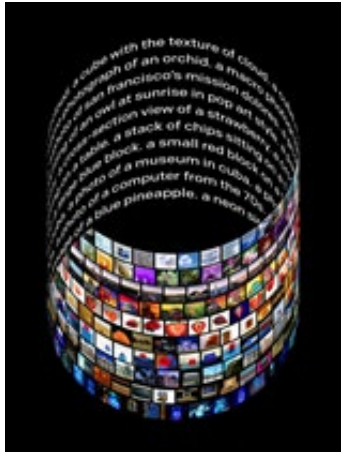
determining an error for the training item using the normalized target output and the normalized output; and

using the error to adjust the current values of the main parameters of the neural network.

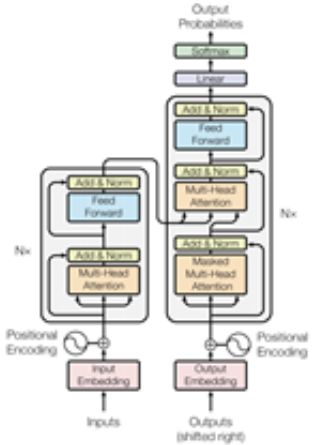
The claim describes the flow of data and the data manipulation process during the training process performed separately from the model inference process.

PI IP LAW

Data Become Much More Important



DALL- E



TRANSFORMER BASED HUGE MODELS

Google does not provide a commercial license on parameters on AlphaFolds2 while releasing the model as an open source.

PI IP LAW

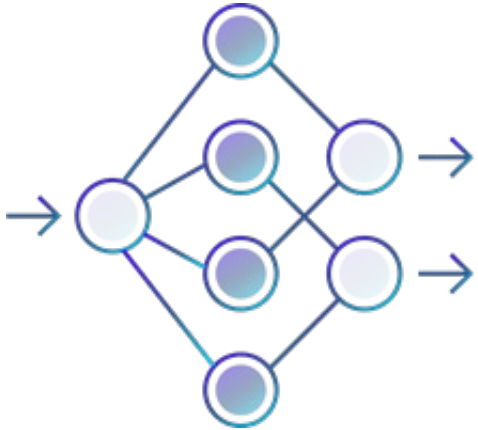
Data Structure Claims for AI Inventions

"'A data structure recorded computer-readable medium' in which the contents of processing performed by the computer are specified due to the recorded data structure, can be claimed as an invention of a device."

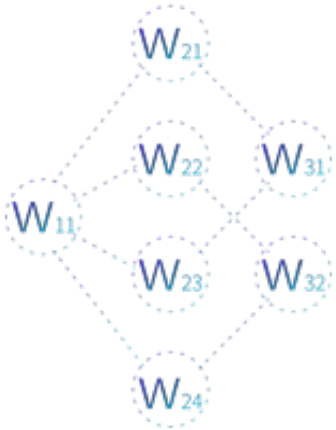
* Korean examination guidelines for computer software

PI IP LAW

Korean IP Office



AI MODELS



STRUCTURED
PARAMETERS

KIPO still does not set a concrete standpoint on whether it will allow such types of claims as a patent - eligible matter

Summary

- Patent Protection is essential to AI innovations as well.
- Claiming patents on AI technologies requires careful consideration of aspects of innovation.
- As machine learning becomes more data-centric, the IP system needs to be improved or even reformed.

PI IP LAW
LEE, PARK & ASSOCIATES

- Address Urbanhive 11F, 476, Gangnam-daero, Gangnam-gu, Seoul, 06120, KOREA
- Email piip@piip.co.kr
- Tel +82 2-6745-0410
- Fax +82 70-4725-0027
- Website www.piip.co.kr