Advances in Artificial Intelligence and Machine Learning; Theoretical Research 2 (3) 456-468 Received 4-09-22; Accepted 24-09-22; Published 30-09-22

Invariance-Based Approach Explains Empirical Formulas from Pavement Engineering to Deep Learning

Edgar Daniel Rodriguez Velasquez edrodriguezvelasquez@miners.utep.edu
Department of Civil Engineering
Universidad de Piura in Peru (UDEP),
Peru
Department of Civil Engineering
University of Texas at El Paso, El Paso, Texas,
USA

Olga Kosheleva olgak@utep.edu
Department of Teacher Education University of Texas at El Paso,
El Paso, Texas,
USA

Vladik Kreinovich vladik@utep.edu
Department of Computer Science University of Texas at El Paso,
El Paso, Texas,
USA

Corresponding Author: Vladik Kreinovich

Abstract

In many application areas, there are effective empirical formulas that need explanation. In this paper, we focus on two such challenges: deep learning, where a so-called softplus activation function is known to be very efficient, and pavement engineering, where there are empirical formulas describing the dependence of the pavement strength on the properties of the underlying soil. We show that similar scale-invariance ideas can explain both types of formulas – and, in the case of pavement engineering, invariance ideas can lead to a new formula that combines the advantages of several known ones.

Keywords: Invariance, Scale-invariance, Shift-invariance, Pavement engineering, Neural network, Softplus activation function.

1. FORMULATION OF THE PAVEMENT ENGINEERING PROBLEM

How resilient modulus depends on the soil suction: an important practical question. The strength of the pavement is affected by the strength of the underlying soil. That strength is described by the resilient modulus M_R – the coefficient that described how the recoverable strain ε_r depends on the cyclic stress σ_d: $\sigma_d = M_R \cdot \varepsilon_r$. The resilient modulus, in its turn, is affected by the amount

Citation: Edgar Daniel Rodriguez Velasquez. Invariance-Based Approach Explains Empirical Formulas from Pavement Engineering to Deep Learning Advances in Artificial Intelligence and Machine Learning. 2022;2(3):30.

of water present in the soil. This amount is described by the soil suction ψ – negative pore-water pressure, i.e., the amount of energy needed to extract a unit of water from the soil.

So, to predict how strong the pavement will be under different soil water saturation conditions, it is necessary to know how the resilient modulus M_R depends on the suction ψ.

What is known about this dependence. At present, there are no from-first-principle theoretical formulas describing the desired dependence, but there are several empirical formulas. These formulas have been summarized and analyzed in a survey paper [1].

According to [1], there are two types of empirical formulas:

- formulas of the type $M_R = c \cdot (1 + a \cdot \psi)^b$ first proposed in [2, 3] and finalized – by taking into account a more accurate description of the parameters a, b, and c on other parameters of the pavement – in [4], and
- formulas of the type $M_R = c + a \cdot \psi^b$ proposed in [5].

As shown in [1], in general, both formulas lead to equally good description of empirical data: in some cases, the first formula provides a better fit, in other cases, the second formula leads to a more accurate description of the data.

Resulting problems. It is desirable to come up with a theoretical explanations for the successful empirical formulas.

Ideally, it is also desirable to use these explanations to come up with a single formula that will hopefully lead to an even better fit with empirical data. This is what we do in this paper.

2. FORMULATION OF THE NEURAL NETWORK PROBLEM

What is a neuron, what is a neural network, and what is deep learning: a brief reminder. A usual computational neuron transform its inputs $x_1, \ldots, x_n$ into the output

$$y = s_0(w_1 \cdot x_1 + \ldots + w_n \cdot x_n - w_0),$$

where w_i are coefficients that need to be adjusted during training, and $s_0(x)$ is a non-linear function knows as an *activation function*.

In a neural network, first, we have neurons that process inputs to the network – i.e., data to be processed. Outputs of these neurons become inputs to other neurons, etc. At the end, we may need to compute a linear combination of the output signals to produce the final result.

In the traditional neural networks, input signals were processes by several non-linear neurons, after which a linear combination of these neurons' outputs was returned as the final result; see, e.g., [6]. In other words, such neural networks contained only one layer of non-linear neurons.

In general in computing, a natural idea is to combine several computational units into a single computational process. This is how software is usually written: by combining software modules. In line with this idea, researchers have proposed to combine one-nonlinear-layer neural networks into a single more powerful computational device, i.e., to design and use neural networks with several non-linear layers. In the last decades, it was shown that such multi-layer neural networks indeed lead to much better computational results; see, e.g., [7]. Such networks are known as *deep* neural networks – and the process of training such a neural network to fit the available patterns is known as *deep learning*.

Which activation function works best? Most traditional one-nonlinear-layer neural networks used the so-called sigmoid activation function

$$s_0(x) = \frac{1}{1+\exp(-x)}.$$

This function was originally proposed because it reasonably accurately described how signals are processed in biological neurons. Later on, it turned out that, in many applications, this activation function indeed works better than many other non-linear functions. This empirical confirmation is not accidental: biological neurons are a product of billions of years of improving evolution, so it is not surprising that their design – including the selection of the corresponding activation function – is practically optimal.

However, when researchers tried to design multi-layer neural networks based on the sigmoid activation function, these networks did not perform very well. The current breakthrough – and the resulting spectacular successes of deep learning – was achieved when researchers tried different activation functions. Most current applications of deep learning use the so-called Rectified Linear Unit (ReLU) activation function $s_0(x) = \max(x, 0)$ (see, e.g., [7]). However, in some practical applications, the ReLU activation function is not perfect. The main reason for this is that this function is not everywhere differentiable – namely, it is not differentiable at $x = 0$. Because of this, the input-output dependence $y(x_1, \ldots, x_n)$ produced by training such a neural network is not everywhere smooth, while smoothness is important in many applications, e.g., in many control applications.

Researchers and practitioners tried many smooth alternatives to ReLU. It turned out that in many different application cases with different configurations of neural networks, the following smooth approximation to ReLU works better: $s_0(x) = \ln(1+\exp(x))$. This activation function is known as *softplus*; see, e.g., [8, 9]. Many possible smooth approximations to ReLU have been tried, so why this one works better is not clear.

In this paper, we explain why this particular activation function is empirically successful.

General comments.

- As promised, the two explanations – for pavement engineering and for neural networks – are based on the same scale invariance idea.
- In the case of pavement engineering, our theoretical analysis does not just explain the existing empirical formulas, it helps to come up with a new formula that combines the advantages

of several empirical formulas. In contrast, for neural networks, we do not (yet) propose any new design or any new usage of a neural network. However, it is well known that in many application areas, a proper theoretical understanding of empirical phenomena was a first step that eventually led to better designs and better techniques – and our analysis of pavement engineering confirms this general trend. We therefore hope that our theoretical explanation will eventually lead to better neural network techniques as well.

3. SCALE-INVARIANCE: GENERAL REMINDER

Possibility of different scales. One of the main objectives of science is to find relations between physical quantities. In the formulas and algorithms, these quantities are represented by numerical values. These values depend not only on the quantity itself, they also depend on the choice of a measuring unit and, for some quantities like temperature or time, on the selection of a starting point.

If we replace the original measuring unit with the one which is λ times smaller, then all numerical values are multiplied by λ: $x \mapsto \lambda \cdot x$. For example, if we replace meters with centimeters, all the numerical values are multiplied by 100, so, e.g., 1.7 m becomes 170 cm. If we replace the original starting point with the one which is x_0 units smaller, then this value x_0 is added to all the numerical values: $x \mapsto x + x_0$.

Scale-invariance. In many physical situations, there is no natural measuring unit, the choice of a measuring unit is a matter of convention. In this case, it makes sense to require that the desired dependence $y = f(x)$ between two physical quantities x and y does not change if we use a different measuring unit for x.

Of course, since the quantities x and y are related, the selection of a different measuring unit for x may necessitate the selection of a different measuring unit for y. For example, the formula $y = x^2$ for the area of a square does not depend on the choice of a measuring unit for measuring length, but if we change the unit of length from meters to centimeters, we will need to appropriately change the unit of area from square meters to square centimeters.

So, in general, scale-invariance means that for every $\lambda > 0$, there exists a value $\mu(\lambda)$ depending on λ for which $y = f(x)$ implies $y' = f(x')$, where $x' = \lambda \cdot x$ and $y' = \mu(\lambda) \cdot x$. It is known (see, e.g., [10]) that for continuous dependencies (and most physical relations are continuous and even differentiable) invariance implies that $y = A \cdot x^B$ for some constants A and B – i.e., that the scale-invariant dependence between x and y is described by a power law.

The above empirical formulas are more complex than the power law, so we need additional ideas.

Taking shift-invariance into account. A natural idea is to take into account that in some situations, we can also select different starting points. Since there is no physically preferable starting point, it makes sense to require that the dependence $y = f(x)$ does not change if we change not only the measuring unit for x, but also the starting point for x, i.e., if we replace x with $x' = \lambda \cdot x + x_0$.

In this case, it makes sense to similarly require that for every pair (λ, x_0) there exists a pair $(\mu(\lambda, x_0), y_0(\lambda, x_0))$ for which $y = f(x)$ implies that $y' = f(x')$, where

$$y' = \mu(\lambda, x_0) \cdot y + y_0(\lambda, x_0).$$

It can be proven that for smooth functions $f(x)$, this scale-and-shift-invariance requirement implies that the corresponding function $f(x)$ is linear.

Comments.

- For readers' convenience, this proof is placed in the special (last but one) section of this paper.
- In general, the fact that a dependence does not change if we apply a certain transformation is known as *invariance*. Invariance is one of the main tools in modern theoretical physics (see, e.g., [11, 12]), to the extent that, starting from quark theory, most theories are proposed not in terms of differential equations – as in Newton's times – but in terms of the corresponding invariances. In this paper, we deal with scale-invariance and shift-invariance, which are among the simplest – although still important – types of invariance.

Indirect dependencies. In some cases, we have a direct dependence $y = f(x)$. In such cases, it makes sense to assume that this dependence is either scale-invariant or scale-and-shift-invariant.

However, in many other situations, the dependence between physical quantities is indirect. For example, it can be that y directly depends on some auxiliary quantity z which, in turn, directly depends on the quantity x. In this case, we have $y = f(z)$ and $z = g(x)$, so $y = f(g(x))$. In this case, the dependence of y on x is a composition of two scale-invariant or scale-and-shift-invariant functions.

It is worth mentioning that a composition of two scale-invariant functions is also scale-invariant, and a composition of two scale-and-shift-invariant functions is also scale-and-shift-invariant. Thus, the only case when we get new functions is when we take a composition of two functions which are differently invariant.

We may have a more complex situation, in which we have two (or even more) intermediate quantities: y depends on z, z depends on t, and t depends on x. In this case, $f(x)$ is a composition of three (or more) invariant functions, e.g., $f(g(h(x)))$. Similarly to the previous case, the only way to get new functions is when every two neighboring functions in this composition are differently invariant.

4. LET US APPLY THESE INVARIANCE IDEAS TO OUR PAVEMENT ENGINEERING PROBLEM

Invariance ideas explain the two empirical formulas. As we have mentioned, neither power laws nor linear functions provide a good approximation to the actual dependence of the resilient modulus

M_R on the soil suction ψ. Since we cannot describe this empirical dependence by a single invariant function, a natural idea is to consider compositions of two invariant functions.

As we have mentioned, to get a composition function different from simply invariant ones, the functions involved in the composition must be differently invariant: one of them must be scale-invariant (i.e., described by the power law), while another one must be scale-and-shift-invariant, i.e., be a linear function. Depending on which of these two functions we apply first, we get two different compositions:

- If we apply a power law first, then we first get $z = A \cdot \psi^b$ for some constants A and b. After that, we apply a linear function $M_R = c + p \cdot z$. Then, the resulting dependence of M_R on ψ has the form $M_R = c + p \cdot A \cdot \psi^b$, i.e., for $a = p \cdot A$, exactly the form $M_R = c + a \cdot \psi^b$, which is one of the empirical formulas.
- On the other hand, let us consider the case when we first apply a linear function. In this case, we get $z = p \cdot \psi + q$, i.e., equivalently, $z = q \cdot (1 + a \cdot \psi)$, where we denoted $a \stackrel{\text{def}}{=} p/q$. After this, we apply a power law $M_R = A \cdot z^b$, so we get $M_R = A \cdot q^b \cdot (1 + a \cdot \psi)^b$, i.e., for $c = A \cdot q^b$, exactly the form $M_R = c \cdot (1 + a \cdot \psi)^b$, which is another of the two empirical formulas.

Thus, both empirical formulas indeed follow from natural invariance requirements.

How can we find a single more accurate formula? As we have mentioned, in some cases, one of the two empirical formulas works better, in other cases, the second formula is more accurate. It is desirable to have a single formula that would combine the advantages of both, i.e., that would be as accurate (or even more accurate) then both empirical formulas.

Based on our analysis, a natural idea is to form the corresponding function as a composition of several invariant ones. We cannot have only two such functions – then all we get is the two above-described empirical formulas. Thus, to get a more accurate description, we need to have a composition of at least three invariant functions. As we have mentioned earlier, in such a composition, neighboring functions should be differently invariant. Thus, depending on whether we start with a scale-invariant function or with a scale-and-shift-invariant function, we get the following two expressions.

If we start with a scale-invariant function, then:

- we first apply a power law followed by a linear transformation; we already know that this leads to $z = c + a \cdot \psi^b$, i.e., equivalently, to $z = c \cdot (1 + a' \cdot \psi^b)$, where $a' \stackrel{\text{def}}{=} a/c$;
- finally, we apply a power law to the result z of the first two steps, resulting in $M_R = A \cdot (c \cdot (1 + a' \cdot \psi^b))^B$, i.e., equivalently, in

$$M_R = A' \cdot (1 + a' \cdot \psi^b)^B, \tag{1}$$

where we denoted $A' \stackrel{\text{def}}{=} A \cdot c^B$.

On the other hand, if we start with a scale-and-shift-invariant function, then:

- we first apply a linear function followed by a power law; we already know that this leads to $z = c \cdot (1 + a \cdot \psi)^b$;
- finally, we apply a linear function to the result z of the first two steps, resulting in $M_R = A \cdot c \cdot (1 + a \cdot \psi)^b + B$, i.e., equivalently, in

$$M_R = A' \cdot (1 + a' \cdot \psi)^b + B, \tag{2}$$

where we denoted $A' \stackrel{\text{def}}{=} A \cdot c$.

Each of the two new formulas (1) and (2) includes both above-described empirical formulas as particular cases. Which of the two new formula is more accurate needs to be determine experimentally; at this point, we do not yet have enough accurate measurements to make a statistically significant choice – but hopefully, in the nearest future, when measurements will become even more accurate, this choice will be done.

5. LET US NOW APPLY INVARIANCE IDEAS TO THE NEURAL NETWORK PROBLEM

What are proper invariances here? Neural networks, in general, deal with signals: they take input signals and they general output signals. In contrast to many physical situations, for signals, in many cases, there *are* preferred units. For example, for intensity of an X-ray image, we can take, as such a measuring unit, an average value of intensity over all healthy people – or, to be more precise, over a sufficiently large sample of healthy people. Thus, in this case, we do not expect scale-invariance.

However, what we do have is shift-invariance. Indeed, for most signals – e.g., for the intensity of an X-ray image – there is usually a background noise level. Researchers and practitioners are well aware of this situation. So, to eliminate the effect of this noise and thus, to deal only with the informative signal, they estimate the amount of the background noise and subtract this estimate from all measured signal values.

The problem is that we cannot find the exact value of this noise – just like we cannot find the exact value of any physical quantity: we can perform more and more accurate measurement, but some uncertainty always remains; see, e.g., [13]. If two groups of researchers come up with two different estimates e and e' to subtract from the original signal X, then the resulting differences $X - e$ and $X - e'$ different by a shift:

$$X - e' = (X - e) + (e - e').$$

Since there is no preferred estimate, all estimates are equally reasonable. So, it makes sense to require that the processing algorithms – in particular, the corresponding activation function – should not change after such shift. In other words, if we consider direct signal-to-signal transformations $y = s_0(x)$, we should – similarly to scale-invariance – require that for every possible x-shift value x_0, there should exist a corresponding y-shift value $y_0(x_0)$ such that $y = s_0(x)$ implies $y' = s_0(x')$, where $x' = x + x_0$ and $y' = y + y_0(x_0)$.

Need for indirect transformations. The above formulation sounds reasonable, but the problem is that this shift-invariance requirement implies that $s_0(x)$ is a linear function; see, e.g., [10]. If we only

use neurons with a linear activation function, we will only be able to represent linear dependencies – which many real-life processes are nonlinear.

Thus, as we have mentioned, in cases when we cannot use a single invariant function – corresponding to direct dependence, we should consider compositions of several invariant functions, i.e., consider the case when the dependence of the output signal y on the input signal x is indirect: e.g., y directly depends on z and z directly depends on x; or, in more complex cases, y depends on z, z depends on t, and t depends on x.

These intermediate cases cannot be signals of the same type as x and y – otherwise, all these dependencies will be linear, and a composition of any number of linear functions is still linear. Thus, it makes sense to require that the intermediate quantities represent values of some quantities – and thus, for their transformations, we should require either scale-invariance or scale-and-shift-invariance. Let us analyze what functions $y = s_0(x)$ we get as a result.

Case of a single intermediate quantity. Let us first consider the case when we have only one intermediate quantity z, i.e., when y directly depends on z, as $y = f(z)$ and z directly depends on x, as $z = g(x)$, so that $y = f(g(x))$.

Since y is shift-invariant and z is scale-invariant, a natural requirement for the dependence $y = f(z)$ is that for every scaling value λ, there should exist a corresponding y-shift value $y_0(\lambda)$ such that $y = f(z)$ implies $y' = f(z')$, where $z' = \lambda \cdot z$ and $y' = y + y_0(\lambda)$. It is known that this property implies that $f(z) = a + b \cdot \ln(z)$ [10].

Similarly, since z is scale-invariant and x is shift-invariant, a natural requirement for the dependence $z = g(x)$ is that for every scaling x-shift x_0, there should exist a corresponding scaling value $\lambda(x_0)$ such that $z = g(x)$ implies $z' = g(x')$, where $z' = \lambda(x_0) \cdot z$ and $x' = x + x_0$. It is known that this property implies that $g(x) = A \cdot \exp(B \cdot x)$ [10].

Thus, the composition $y = f(g(x))$ has the form

$$y = a + b \cdot \ln(A \cdot \exp(B \cdot x)) = a + b \cdot (\ln(A) + B \cdot x).$$

So, we again get a linear function $y = p \cdot x + q$, where $p = b \cdot B$ and $q = a + b \cdot \ln(A)$, and this is exactly what we wanted to avoid.

We need at least two intermediate quantities. So, we cannot have a composition of just two functions, we need a composition of at least three functions $s_0(x) = f(h(g(x)))$, where:

- the function $f(z)$ describes the dependence of the output signal y on the first intermediate quantity z,
- the function $h(t)$ describes how the first intermediate quantity z depends on the second intermediate quantity t, and
- the function $g(x)$ describes how the second intermediate quantity t depends on the inputs signal x.

We already know that $f(z)$ has the form $f(z) = a + b \cdot \ln(z)$ and that $g(x)$ has the form $g(x) = A \cdot \exp(B \cdot x)$. We also known, from our previous analysis, that the dependence $h(t)$ must be either scale-invariant or scale-and-shift invariant, i.e., it should be described either by a power law $s = p \cdot t^q$ or by a linear function $s = p \cdot t + q$. Let us consider both possibilities one by one.

First possibility, when the dependence $h(t)$ is scale-invariant. In this case,

$$z = h(g(x)) = p \cdot (A \cdot \exp(B \cdot x))^q = A' \cdot \exp(B' \cdot x),$$

where $A' = p \cdot A^q$ and $B' = B \cdot q$. This is the same form as in the case where there is only one intermediate quantity, so the resulting function $s_0(x) = f(g(h(x)))$ is linear. Thus, this possibility does not lead to any non-linear activation function either.

Second possibility, when the dependence $z = h(t)$ is scale-and-shift-invariant. In this case,

$$z = h(g(x)) = p \cdot A \cdot \exp(B \cdot x) + q,$$

i.e., equivalently, $z = q \cdot (1 + p' \cdot \exp(B \cdot x))$, where $p' = (p \cdot A)/q$. Here, $p' = \exp(p'')$, where $p'' = \ln(p')$, thus $z = q \cdot (1 + \exp(B \cdot x + p''))$. Hence, for $y = f(z)$, we have

$$y = a + b \cdot \ln(q \cdot (1 + \exp(B \cdot x + p''))),$$

so $y = a + b \cdot \ln(q) + \ln(1 + \exp(B \cdot x + p''))$, i.e.,

$$y = a' + b \cdot (1 + \exp(B \cdot x + p'')),$$

where $a' = a + b \cdot \ln(q)$. The resulting expression is exactly softplus; to be more precise, it can be obtained if we:

- first, apply a linear transformation $x \mapsto x' = B \cdot x + p''$,
- then apply softplus to x', resulting in

  $$y' = \ln(1 + \exp(x')),$$

  and
- finally, apply, to y', a linear transformation $y' \to a' + b \cdot y'$.

As we have mentioned, in a neural network, linear transformations are allowed. So, a linear transformation of an activation function does not change the class of function that can be computed on any multi-layer configuration of such neurons. Thus, we can conclude that the invariance requirement uniquely determines the softplus activation function – so the empirical successes of this function can indeed be explained by natural invariance requirements.

6. PROOF

Definition 1. *We say that a smooth (differentiable) function $f(x)$ is* scale-and-shift-invariant *if for every pair (λ, x_0) there exists a pair $(\mu(\lambda, x_0), y_0(\lambda, x_0))$ for which $y = f(x)$ implies that $y' = f(x')$, where $y' = \mu(\lambda, x_0) \cdot y + y_0(\lambda, x_0)$.*

Proposition 1. *A function $f(x)$ is scale-and-shift-invariant if and only if it is linear, i.e., if it has the form $f(x) = a \cdot x + b$ for some a and b.*

Proof.

1°. It is easy to show that linear functions are scale-and-shift-invariant in the sense of Definition 1. Thus, to complete the proof, it is sufficient to show that every scale-and-shift-invariant function is linear.

Indeed, let us assume that $f(x)$ is scale-and-shift-invariant.

2°. Let us prove that this invariance implies that the function $f(x)$ satisfies a certain functional equation.

Indeed, for $\lambda = 1$, the requirement from Definition 1 means that $y = f(x)$ implies

$$y' = f(x + x_0) = \mu(1, x_0) \cdot y + y_0(1, x_0). \tag{3}$$

Since $y = f(x)$, this means that

$$f(x + x_0) = \mu(1, x_0) \cdot f(x) + y_0(1, x_0). \tag{4}$$

3°. Let us prove that the functions $\mu(1, x_0)$ and $y_0(1, x_0)$ are differentiable.

Indeed, the equality (4) is true for all possible value x. In particular, we can select two different values $x_1 \neq x_2$ and conclude that

$$f(x_1 + x_0) = \mu(1, x_0) \cdot f(x_1) + y_0(1, x_0) \text{ and}$$
$$f(x_2 + x_0) = \mu(1, x_0) \cdot f(x_2) + y_0(1, x_0). \tag{5}$$

Thus, for each x_0, we have two linear equations with constant coefficients to determine two unknowns $\mu(1, x_0)$ and $y_0(1, x_0)$.

A solution to such a system of linear equations is a linear combination of the right-hand sides, i.e., in this case, a linear combination of $f(x_1+x_0)$ and $f(x_2+x_0)$. Since the function $f(x)$ is differentiable, each such linear combination is differentiable too.

The statement is proven.

4°. Let us prove that the scale-and-shift-invariance requirement leads to a differential equation for $f(x)$.

Indeed, based on Definition 1 and on Part 3 of this proof, we conclude that all three functions $f(x)$, $\mu(1, x_0)$ and $y_0(1, x_0)$ are differentiable. Thus, we can differentiate both sides of the formula (4) with respect to x_0 and get

$$\frac{df}{dx}(x + x_0) = \frac{\partial \mu(1, x_0)}{\partial x_0} \cdot f(x) + \frac{\partial y_0(1, x_0)}{\partial x_0}. \tag{6}$$

In particular, for $x_0 = 0$, we get

$$\frac{df(x)}{dx} = p \cdot f(x) + q, \tag{7}$$

where we denoted

$$p \stackrel{\text{def}}{=} \frac{\partial \mu(1, x_0)}{\partial x_0}_{|x_0=0} \text{ and } p \stackrel{\text{def}}{=} \frac{\partial y_0(1, x_0)}{\partial x_0}_{|x_0=0}. \tag{8}$$

5°. Let us now find all the solutions to the differential equation (7) and check which of them are invariant with respect to scalings $x \mapsto \lambda \cdot x$ as well.

The form of the solution depends on whether the coefficient p is equal to 0 or not.

5.1°. If $p = 0$, then $f(x)$ is a linear function.

5.2°. Let us now consider the case when $p \neq 0$.

When $p \neq 0$, we can separate the variables if we divide both sides by $p \cdot f + q$ and multiply both sides by dx. This way, we get

$$\frac{df}{p \cdot f + q} = dx. \tag{9}$$

The left-hand side of this equality can be re-formulated as

$$\frac{df}{p \cdot f + q} = \frac{1}{p} \cdot \frac{df}{f + \frac{q}{p}} = \frac{1}{p} \cdot \frac{d\left(f + \frac{q}{p}\right)}{f + \frac{q}{p}}; \tag{10}$$

Thus, the equality (9) takes the form

$$\frac{1}{p} \cdot \frac{d\left(f + \frac{q}{p}\right)}{f + \frac{q}{p}} = dx. \tag{11}$$

Integrating both sides, we get

$$\frac{1}{p} \cdot \ln\left(f + \frac{q}{p}\right) = x + C, \tag{12}$$

where C is the integration constant. Thus,

$$\ln(f + c) = p \cdot x + d, \tag{13}$$

where we denoted

$$c \stackrel{\text{def}}{=} \frac{q}{p} \text{ and } d \stackrel{\text{def}}{=} p \cdot C. \tag{14}$$

By applying $\exp$ to both sides of the equality (11), we get

$$f(x) + c = \exp(p \cdot x + d), \tag{15}$$

so

$$f(x) = \exp(p \cdot x + d) - c. \tag{16}$$

For any $\lambda > 1$, the function

$$f(\lambda \cdot x) = \exp(p \cdot \lambda \cdot x + d) - c \tag{15}$$

grows much faster than $f(x)$: for $x \to \infty$ if $p > 0$ and for $x \to -\infty$ if $p < 0$. Thus, the function $f(\lambda \cdot x)$ cannot be equal to $\mu(\lambda, 0) \cdot f(x) + y_0(\lambda, 0)$. Hence, for $p \neq 0$, we cannot have a scale-and-shift-invariant function $f(x)$.

So, we conclude that $p = 0$ and thus, the function $f(x)$ is linear. The proposition is proven.

7. CONCLUSION

This paper attacks an important problem: to find a theoretical explanation for empirical formulas and techniques and, ideally, to use this explanation to come up with more accurate formulas and more effective techniques. As case studies, we use empirical formulas from pavement engineering and from deep learning. It turns out that the same physics-based mathematical apparatus of invariances can explain formulas from both application areas. In the pavement engineering example, we not only provide a theoretical explanation for several empirical formulas, we also show that invariance techniques lead to a new formula that combined the advantages of several known ones.

In the deep learning example, we explain the empirical success of the sofplus activation function – the most empirically effective smooth alternative to the usual Rectified Linear Unit (ReLU) activation function. While, at present, we simply provide a theoretical explanation for an empirically successful formula, we hope that in the future – similarly to the pavement engineering case – such invariance-based analysis will lead to the design of new, even more effective, deep learning techniques.

8. ACKNOWLEDGEMENT

This work was supported in part by the National Science Foundation grants 1623190 (A Model of Change for Preparing a New Generation for Professional Practice in Computer Science), and HRD-1834620 and HRD-2034030 (CAHSI Includes), and by the AT&T Fellowship in Information Technology.

It was also supported by the program of the development of the Scientific-Educational Mathematical Center of Volga Federal District No. 075-02-2020-1478, and by a grant from the Hungarian National Research, Development and Innovation Office (NRDI).

The authors are greatly thankful to the anonymous referees for valuable suggestions.

References

[1] Han Z, Vanapalli SK, "State-Of-The-Art: Prediction of Resilient Modulus Of Unsaturated Subgrade Soils". International Journal of Geomechanics. 2016;16:04015104.

[2] Liang RY, Rabab'ah S, Khasawneh M. "Predicting Moisture-Dependent Resilient Modulus of Cohesive Soils Using Soil Suction Concept". Journal of Transportation Engineering. 2008;134:34-40.

[3] Yang SR, Huang WH, Tai YT. "Variation of Resilient Modulus With Soil Suction for Compacted Subgrade Soils", Transportation Research Record. Journal of the Transportation Research Board. 2005;1903:99-106.

[4] Ng CWW, Zhou C, Yuan Q, Xu J. "Resilient Modulus Of Unsaturated Subgrade Soil: Experimental and Theoretical Investigations". Canadian Geotechnical Journal. 2013;50:223-232.

[5] Khoury NN, Brooks R, Khoury C. N. "Environmental Influences On the Engineering Behavior of Unsaturated Undisturbed Subgrade Soils: Effect of Soil Suctions on Resilient Modulus". International Journal of Geotechnical Engineering. 2009;3:303-311.

[6] Bishop CM. Pattern Recognition and Machine Learning, Springer, New York. 2006.

[7] Goodfellow I, Bengio Y, Courville A. Deep Learning. Cambridge, Massachusetts: MIT Press. 2016.

[8] Dugas C, Bengio Y, Bélisle F, Nadeau C, Garcia R. "Incorporating Second-Order Functional Knowledge for Better Option Pricing", Proceedings of the 13th International Conference on Neural Information Processing Systems NIPS'00, Denver, Colorado, November 27-30, 2000, MIT Press. 2020:451-457.

[9] Glorot X, Bordes A, Bengio Y. "Deep Sparse Rectifier Neural Networks", Proceedings of the International Conference on Artificial Intelligence and Statistics, Ft. Lauderdale, Florida. 2011:315-323.

[10] Rodriguez Velasquez ED, Kreinovich V, Kosheleva O. "Invariance-Based Approach: General Methods and Pavement Engineering Case Study".International Journal of General Systems. 2021;50:672-702.

[11] Feynman R, Leighton R, Sands M. The Feynman Lectures on Physics, Addison Wesley, Boston, Massachusetts, 2005.

[12] Thorne KS, Blandford RD. Modern Classical Physics: Optics, Fluids, Plasmas, Elasticity, Relativity, and Statistical Physics, Princeton University Press, Princeton, New Jersey, 2017.

[13] Rabinovich SG. Measurement Errors and Uncertainty: Theory and Practice. Springer Verlag, New York, 2005.